DOCUMENT RESUME

ED 175 447 IR 007 614

TITLE Mission of the Future. Proceedings of the Annual Convention of the Association for the Development of Computer-Based Instructional Systems. Volume I: General Session Papers and Project Reports (San Diego, California, February 27 to March 1, 1979).

INSTITUTION Association for the Development of Computer-based Instructional Systems.

PUB DATE Mar 79

NOTE 332p.; Legibility varies; For related documents, see IR 007 614-616

AVAILABLE FROM Western Washington University, Bellingham, Washington 98225 (3 Volumes, $30.00)

EDRS PRICE MF01 Plus Postage. PC Not Available from EDRS.

DESCRIPTORS *Computer Assisted Instruction; Computer Managed Instruction; Educational Technology; Elementary Secondary Education; Interaction; Learning Theories; Man Machine Systems; Medical Education; Models; Postsecondary Education; *Simulation; *Videodisc Recordings

IDENTIFIERS BASIC; COBOL; Minicomputers; PLATO

ABSTRACT

The first of three volumes of papers presented at the 1979 ADCIS convention, this collection includes most of the 30 papers presented at the general session and 34 project reports. The general session opened with seven papers on various aspects of videodisc technology and five papers describing phases in the life of CBE sites. The remaining presentations were concerned with a variety of topics, such as the use of a speech synthesizer with CBI, intelligent instruction systems, learning theory, evaluation, training teachers and administrators in computer applications, learner interaction in CAI, course development, authoring, and the application of bibliographic information retrieval techniques to CBE materials; a report on the development and implementation of CICERO at the Open University for distance learning is included. Projects described in brief reports include a university educational technology center, simulations for medical training, business applications, providing feedback in very large classes, CAI for the deaf, and CAI at several educational levels. Some papers include brief bibliographies. (RAO)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

U.S. DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
NATIONAL INSTITUTE OF
EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

ED175447

MISSION OF THE FUTURE

PROCEEDINGS OF THE ANNUAL CONVENTION

OF THE

ASSOCIATION FOR THE DEVELOPMENT OF COMPUTER-BASED INSTRUCTIONAL SYSTEMS

VOLUME I

GENERAL SESSION PAPERS AND PROJECT REPORTS

1979

ISLANDIA HYATT HOUSE

SAN DIEGO, CALIFORNIA

FEBRUARY 27 TO MARCH 1

"PERMISSION TO REPRODUCE THIS MATERIAL IN MICROFICHE ONLY HAS BEEN GRANTED BY
Joan Lauer Hayes

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

IR007614

VOLUME I: GENERAL SESSION PAPERS AND PROJECT REPORTS

TABLE OF CONTENTS

I. GENERAL SESSION

"Instruction for the Future--The Computer Video Interface"
Richard Pollack 1

"Instruction for the Future--The Computer Video Interface"
Alfred Bork 5

"Instruction for Future--The Computer Video Interface"
Paul Lorton 9-1

(Title Unknown)
Ira P. Goldstein 9-2

"Videodisc Overview"
J. D. Fletcher 10

"The Impact of Videodisc on Instructional Systems Development"
William W. Hawkins 23

"Intelligent Videodisc and the Learning Society"
Andrew Molnar 32-1

"Five Phases in the Life of CBE Sites: Consideration of Alternatives - I. Planning the Project"
Lynn Misselt 32-2

"Five Phases in the Life of CBE Sites: Consideration of Alternatives - II. Staff Selection and Retention"
Larry Francis 33

"Five Phases in the Life of CBE Sites: Consideration of Alternatives - III. Lesson Development."
Eileen Call-Himwich 40

"Five Phases in the Life of CBE Sites: Consideration of Alternatives - IV. Implementation and Site Management"
Alec Himwich 46

"Five Phases in the Life of CBE Sites: Consideration of Alternatives - V. Results and Final Evaluations"
R. A. Avner 53

"The Use of MISS in Computer-Based Instruction"
W. R. Sanders, A. Levine and R. Laddaga 58

"Intelligent Instructional Systems in Training"
J. D. Fletcher and Frank Zdybel 68

"The Implications of a Formal Model of the Syllabus for a Theory of learning"
Ira P. Goldstein 84

"Towards the Intelligent Evaluation of Complex Computer Based Learning Environments"
Guy Groen 96

"Socratic Teaching of Causal Knowledge and Reasoning"
Madeleine Bates, Geoffrey Brown, and Allan Collins 104

"Toward the Development of Intelligent CAI"
William Bregar and Arthur Farley 120

"What Should the Future Instructional Theory for CBI Look Like?"
Greg P. Kearsley 129

"A Preliminary Report on the Development and Application of CICERO, an Interactive and Batch Computer Assisted/Managed Learning System Developed for Use in the Distance Teaching Environment"
Aldwyn J. R. Cooper 140

"Learning with a Computer: One Year of PLATO in a European University"
Bert Camstra 145-1

"How Should We Train Teachers and Administrators in Computer Application to Education?"
Michael Mocciola 146

"Toward A Computer Coach for Elementary Programming"
Mark L. Miller 156

"Cost Effective Implementation of Music Notation on CRT Terminals"
Richard L. Hamilton and Rosemary N. Killam 183

"Learner Interaction in Computer-Assisted Instruction"
Esther R. Steinberg 194

"Computer Based Education as a Learning Tool for Foreign Language Students"
Heimy F. Taylor 203

"Is TCI the Answer to Educational Problems"
Carole A. Bagley, Tom Hansen and Daniel L. Klassen . . . 124

"CAI at the University of Akron: Course Development"
Thomas D. Warner 219

"Outstanding Large-Scale Projects in Business and Education"
Harold F. Rahmlow 236

"Computer Assisted Instruction in an Everyday School Environment"
Deborah A. Cheves and Terry E. Parks 241

"Application of Bibliographic Information Retrieval Techniques to Computer-Based Education Materials"
John S. Edwards and Pier Paolo Pisani 250

II. PROJECT REPORTS

"The Educational Technology Center at Irvine"
Alfred Bork 263

"Development of an Evaluative Computer Simulation for Assessment of Physician Skills in the Diagnosis and Management of Patients with Myocardial Infarction"
Carl W. White and Lisa R. Ehrlich 264

"Providing Detailed, Informative Feedback to Very Large Classes"
C. Michael Levy 265

"COUNSAID: A Computer Aided Achievement Analysis System"
Jeffrey M. Baker 266

"DEMO-GRAPHICS: Teaching Population Dynamics in a Multidisciplinary Framework"
Paul Handler 267

"A Commitment to Education: Basic Math and CAI at the Aetna Life & Casualty"
Natalie N. Lowe 268

"An Evaluation of CAI and Traditional Methods of Improving Computational Skills in Mathematics"
Preston B. Prigger 269

"The Sales Call Simulator: A System for Rapid Production of Specialized Training Courseware"
Silas S. Warner 270

"The PSO TUTOR Training Program"
Celia Davis 271

"Computer Assisted Dental Simulations"
Laurette F. Lipson 272

"An Author Training Course for Military Authors of Instructional Materials"
Michael Szabo 273

"Computer Assisted Instruction For Nursing Education?"
Patricia J. Morin and Julia A. Fisco 277

"Grading Utility System Manager (GUSMAN)"
Herbert Bielawa 278

"NSF CAUSE Contribution to Computing in Natural and Social Sciences at Muskingum"
James L. Smith and Larry Zettel 279

"Description of the Physiology Self-Evaluation Units (PHYSEU) Management and Reporting System at The Ohio State University College of Medicine"
Verl E. Dennis 280

"Computerized Patient Management Simulations in Pediatric Certification"
Ernest N. Skakun 281

"PLATO-Assisted Tactile Speech: Data from Field Evaluations"
Frank A. Saunders, Theresa A. Easley and William Hill . . 282

"Design and Implementation of a Large Scale CMI Project on a Microcomputer System"
Harry W. Alderman, II 283

"An Alternative Curriculum for Computer Literacy Development"
Stewart A. Denenberg 284

"Description of a Computer Based Instructional TV System: DAVID"
James R. vonFeldt 285

"The Qualitative Optimization of Drill and Practice Programs for the Deaf"
Greg P. Kearsley 286

"Computerizing the Institute of Aviation at the University of Illinois"
Stanley R. Trollip 287

"CORK"
Robert J. Foster and Duane Stevens 288

"The Development of a Competency-Based Education Program in Music Theory"
Michael Arenson 289

"College Grade Information Computer System"
Ira L. Karp and Laenu A. G. Karp 290

"Computer-Based Educational Technology Program at the Lister Hill National Center for Biomedical Communications"
Charles M. Goldstein and William H. Ford, Jr. 291

"Self-Paced Computer Curriculum Project - BASIC and COBOL"
David L. Feinstein and Jean L. Feinstein 310

"Experiment in Remoted Instruction for the Federal Aviation Administration Department of Transportation"
Thelma Stacy and Craig Burson 311

"QUICK/FTU: A Quasi-Interactive Computer-Based Instructional System at Florida Technological University"
Larry K. Cottrell 312

"The Development and Utilization of Computer-Assisted Instruction in an Activity Based Introductory College Level Science Course"
Martin F. Rudnick 313

"Microcomputer Construction in High School"
Paul Sadler 314

"A Computer-Assisted Computer Course for Elementary and Middle-School Students"
Norman T. Bell and Lee Droegemueller 315

"CAI in Pharmacology at the University of Alberta"
Louis Pagliaro, Charles Bidwell, David Cook, and Charles Heath 316

"Why Computer Commitment Varies Among High School Teachers"
Ronald Anderson, Thomas Hanson, David Johnson and Daniel Klassen 317

VOLUME II: SPECIAL INTEREST GROUP SESSION PAPERS

I. COMPUTER BASED TRAINING SPECIAL INTEREST GROUP

"Dynamic Military Engagements Through Computer Supported Graphics:
W. M. Darden and S. G. Parrish 318

"Requirements Analysis and Feasibility Study of Computer-Based Instruction for the Federal Aviation Administration Department of Transportation"
John A. Buck, Duane R. Bullard and Theodore B. Fagan . . 331

"Computer-Based Graphics and Instructions for Map Reading Training"
Zita M. Simutis 339

"Naval Tactical Actions War Game (NAVTAG)"
Ronald E. Ladue and Ronald Lambert 347

"A Computer-Based Study Guide for Aviation Certification"
Stanley R. Trollip 376-1

"Results of One Year of Computer-Based Instruction at the United Airlines Flight Operations Training Center"
Charles Buchanan 377

"Evaluating the Use of Interactive, Computer-Managed Instruction to Control the Quality of Self-Paced Training Without Requiring the Presence of an Instructor"
Jesse M. Heines 394

"An Analysis of Instructional Delivery Systems in Vocational Education Comparing Computer-Managed Instruction, Teacher-Delivered Module (Paper), and Lecture-Demonstration for the Same Instructional Module"
Edward Boas, Jr. 409

II. EDUCATORS FOR THE DEAF

"A Generative Computer System to Teach Language to the Deaf:
Madeleine Bates and Kirk Wilson 413

"A Computer-Based Writing Laboratory for Deaf Students"
E. Paul Goldenberg and Leo D. Geoffrion 425

"Access to CAI for the Blind and Visually Handicapped"
Peter Maggs 436

"Real Time Captioning"
Emily McCoy and Robert Shumway 448

"Teaching Language/Communication Skills to Deaf Children
by Means of CAI"
Barbara Peterson, Gary Galbraith, Curt Miller, Dick Butkus
Sharon Borthwick and Robert Lennan 467

"Computer-Based Experiments Using the von Restorff
Phenomenon to Teach Mentally Retarded Persons"
John J. Winters, Jr., and David L. Hoats 471

"Overview of CAI at the National Institute for the Deaf"
James A. vonFeldt 475-1

"Media Development Project for the Hearing Impaired
Videodisc Project"
George Propp 476

"PLATO Computer-Assisted Instruction Applications in
Special Education:
Robert M. Yeager and Vi A. Mullin 478

"Computer Education for the Deaf in the San Francisco
Bay Area"
Geoff Zawolkow 490

III. ELEMENTARY/SECONDARY/JUNIOR COLLEGE INTEREST GROUP

"Developing an Elementary School Problem Solving
Curriculum in CAI"
Nel Noddings 492

"Innovative Classroom Applications of Computer Technology"
James L. Morrison 506

"A Teacher-Controlled, Computer-Assisted Instructional
System for Special Education"
Alice Chiang, Carolyn Stauffer, and Alex Cannara 514

"Instructional Strategies for Vocabulary Development in
the Context of a Prescriptive Model"
Karen K. Block, Dan Frezza, John Greeno and Carol Solomon 531

"The American High School Student and the Computer: A Survey of Attitudes"
Justine C. Baker 560

"Decision-Making in a Large Public School System"
Catherine E. Morgan 584

"CONTEXT: A Challenge for the Language Arts CAI Developer"
John G. Allee, Jr., and Robert L. Williams 599

"Project TELEMATH"
Bud Sanders 610

"The Plato Basic Skills Learning System: Good News for Functionally Illiterate Adults"
Patricia M. Mullinix, Peter Rizza and Ralph Heimer 612

"A Multi-Media Approach to a Survey Course: Information Processing and the Computer"
Sallyann Z. Hanson 619

IV. HEALTH INTEREST GROUP

"An Evaluation of Computer-Based Clinical Case Histories in Health Sciences Education"
Stephen Kornberg 630

"An Instructional Design Technique for Developing Computer Simulations in Dentistry"
Laurette F. Lipson, Gary L. Jones and Thomas Ahasic 640

"A Computer-Based Student Appraisal System: Automated from Sign-On to Item Analysis"
William E. Sorlie, Diane Essex, and John Shatzer 652

"CMI Dental Anatomy"
Gary L. Jones and Dianna Newman 667

"A Study to Compare the Effectiveness of Two CAI Formats and the Subsequent Correlation of Individual Learning Style to Performance"
James Magero 676

"PHYSEU - A System of Learning Medical Physiology"
J. A. Lipsky, J. C. Curry, M. N. Nishikawara, H. P. Pieper, J. A. Rall, and L. L. Gabel 683

"MedPLATO: Sharing Computer-Based Medical Education"
Lisa Brenner and Allan Levy 685

"Computerized Test Production in the Medical Field:
Ernest N. Skakun and Milton W. Petruk 705

"Application and Testing of Computer-Aided Instruction in Pharmacology"
Louis A. Pagliaro and Alan Burkhalter 711

VOLUME III: SPECIAL INTEREST GROUP SESSION PAPERS

V. IMPLEMENTATION INTEREST GROUP

"Faculty Reactions to Implementing of CAI Tutorials: What Can Happen and What to Expect"
Robert E. Burns and William I. Davisson 728

"Computer-Aided Review Lessons in English Composition"
Edward A. Kline 738

"Thematic Information Text Revisions and Adjunct Questions"
M. E. Fetler 753

"SPI for SPSS"
Jeffrey W. Wides 773

"Group Problem-Solving with Automatic Mediation"
David S. Stodolsky 781

"The Instructional Effects of Charting Student Performance During Computer Managed Instruction"
Nick Van Matre, H. S. Pennypacker, William M. Hartman, Bryan E. Brett and Linda O. Ward 795

"Linear Programmed Instruction Versus Computer-Assisted Instruction: A Comparison and Evaluation of Achievement in Comma Usage Among Selected Post-Secondary Business Communication Students"
James D. Bell and Timothy D. Taylor 800

"Computers and Educational Institutions"
Alfred Bork 804

"Relationship of College Students' Study Skills and Locus of Control Orientation with Achievement in a Self-Paced - Computer Managed Course of Instruction"
G. J. Blumenfeld, T. Taylor, I. Newman and A. Johnson . . . 811

"The Development of General Multivariate Software:
J. Jack McArdle 824

"The Demise of the Programmer"
James Schuyler 863

VI. MINICOMPUTER USERS INTEREST GROUP

"Effect of Criterion Manipulation on Time Taken in Self-Paced Learning"
Richard M. Evans and John C. Johns 870

"Design and Implementation of a Shared CAI Control Program on a Mini-Based System"
Barry Hills and Emile Attala 883

"A Micro-Processor Based System as a Classroom Teaching Tool"
Samuel W. Spero 894

"Champagne Quality CAI on a Beer Budget"
George Gerhold and Larry Kheriaty 901

"Teachers Taking a Byte Out of an Apple"
Earl Keyser 909

"Beyond Independence: A View of Games and Personal Computing"
Kenneth L. Modesitt 911

VII: NATIONAL CONSORTIUM FOR COMPUTER BASED MUSIC INSTRUCTION

"Courseware Development for Micro-Computer Based Education in Music"
G. David Peters 922

"Project CLEF: CAI in Music Theory (Update 1979)"
Mary Lou Hultberg, Warren E. Hultberg and Ted Tenny 928

"Controlled Evaluation of a Competency-Based Approach to Teaching Aural Interval Identification:
Fred Hofstetter 935

"Computer-Based Ear Training Instruction for Non-Music Majors"
Michael Arenson 949

VIII. PLATO USERS INTEREST GROUP

"Sex Education on PLATO"
Charles D. Nelson, Patricia Lavanhar, and George Pepper . . 959

"PLATO Simulation for Process Operator Training"
Tom Rebstock and John C. Harkey 976

"PLATO in the Teaching of Foundation Visual Design"
Raymond D. Nichols and Charles Wickham 986

"One Thousand Terminals Produce a Lot of
System Statistics"
Paul Tenezar 990-1

"CAI for Adult Basic Skills Training: Two Applications"
Martin A. Siegel and Zita M. Simutis 991

"PLATO Facilitation of Precision Motion Analysis in
Biomechanics"
David A. Barlow, A. Stuart Markham, Jr., and James G.
Richards 1005

"AIDS: The User's Guide to PLATO"
James Kraatz 1013

"The PLATO Display in the Teaching of Optical Letterspacing"
Raymond D. Nichols 1022

"Microprocessor Delivery of PLATO Courseware"
John Stifle, Stanley Smith and David Andersen 1027

I. General Session Papers

(In order of presentation)

INSTRUCTION FOR THE FUTURE -- THE COMPUTER VIDEO INTERFACE AT THE UNITED STATES NAVAL ACADEMY

by

Richard A. Pollak
Associate Director for
Educational Development
Academic Computing Center
United States Naval Academy
Annapolis, Maryland 21402

At the United States Naval Academy a Computer-Augmented Video Education (CAVE) project has been in existence for over 18 months. This project combines the unique instructional advantages of educational television and computing systems. The interactive powers of the computer have been interfaced with the audio-pictorial capabilities of educational television to form a powerful instructional resource. The Naval Academy's faculty has profited from such an instructional resource because of its potential to automate individualized extra instruction, an instructional concept deeply embeded in the Academy's teaching philosophy. The efficient and innovative use of available educational technology, through CAVE, has proved important to the continuation of this policy in a time of increasing faculty workload and more stringent budget requirements.

The major underlying guideline in combining the two separate forms of educational technology was to use existing commercial hardware at the lowest cost. The Academy already possessed many of the components necessary for the CAVE project. There are over sixty 3/4 inch video playback cassette units in use at the Academy. A specialized room in the library is comprised of sixty carrels, sixteen of which have built-in TV monitors and video cassette players. The Academic Computing Center, a separate organization, operates a Honeywell 6060 computer under the Dartmouth Time-Sharing System (DTSS). There are 256 terminal access ports into the computer which support over 350 interactive terminals of one form or another (over 40 are vector graphic terminals). The learning stations designed for the CAVE project were developed utilizing this existing hardware. The illustration shows the learning stations located in the audio-visual room in the library.

In the summer of 1978, development of a control interface was begun that would allow the computer complete control of the video cassette player. This interface has the potential to: 1) GOTO a specific video player counter number, 2) play a specific counter number, stop the video player and then return to the computer, 3) stop, 4) rewind, and 5) pause. It is hoped that this enhancement will allow for many instructional capabilities which were too complicated to do with the existing hardware configuration, such as: more sophisticated branching between the computer and video, and personalization of the video presentation in accordance with the needs of individual students.

The benefits from the CAVE project were twofold. First, the Naval Academy developed expertise among its faculty and technical support people in the process of combining the two distinct media into one instructional resource and in implementing the best use of this resource. Secondly, the students benefited, because they were exposed to: 1) visual learning experiences while taking an active role in the learning process, 2) remedial, enrichment, and prelaboratory instruction tailored to their individual needs, 3) material that may otherwise have been impractical, time-consuming, or too involved for classroom presentation, 4) theory and practice presented in terms of both concrete examples and pictures, 5) results which were fed back almost instantaneously to both students and instructor describing the students' learning experiences.

Packages have been developed which take into account the full capabilities of the new learning resource. These packages use various modes of presentation including tutorial, laboratory preparation, drill/practice, and simulation. The project is multidisciplinary and presently is represented by chemistry, physics, and economics. The project members are: Frank J. Gomba, Chemistry Department; Rae Jean B. Goodman, Economics Department; Billie Joe Graham, Physics Department; John V. Prestia, Chemistry Department,

M. B. Sousa, (CAPT, USN), Assistant Dean for Educational Resources and Director of Computer Services; and J. C. Thompson, Director, Educational Resources Center.

Appreciation goes to all the "cave dwellers" for many hours of dedicated work that made the project possible. Special recognition is due James W. Schwab, James A. Kiser, and Al Wiess whose technical support and many hours of hard work made the project successful.

Instruction for the Future - The Computer Video Interface

Alfred Bork
Educational Technology Center
University of California
Irvine, California 92717
(714-833-6911)

In recent years, after a long experimental period, we are beginning to understand the role of the computer in creating a humanized, individualized, interactive, learning environment for students. As part of this development we have learned more effective use of graphic capability.

Yet the full range of multimedia capabilities is still not generally available. At least two types of objections apply to many of the solutions to the "computer plus media" problem that have have been proposed so far. First, many of these combinations are technologically clumsy, involving adding devices or other components to the existing computer system in ways that are not elegant solutions to the problem. The second objection has to do with logistic problems. Many of the proposed solutions involve loading various types of devices--one for the computer code, one for the slides, one for the film, and one for the audio. While this type of logistic problem can be overcome experimentally, it presents severe problems when materials are to be used in a widespread way in many different environments.

The Intelligent Videodisc System

Almost five years ago we first began to hear about a new technology which furnishes a convenient and elegant solution to the computer plus multimedia problem just discussed. This technology is the optical videodisc player and the associated videodiscs. Primary developers have been Thomson CSF, Phillips, and MCA. Other companies are rumored to be engaged also.

The videodisc is a convenient method for storing large quantities of information. It is not a magnetic storage medium, as with many computer-based, storage systems, but rather stores in a way similar to phonograph records in track-like environments. In most of the current discs the material is impressed on the disc. Another technique which has been explored is the photographic technique, where the grooves are photographic reproductions on a very fine grain film.

The coding on these tracks is read by a laser or other light source. The tracks each contain an address, and so it is possible to go to a particular track without difficulty. In most of the versions of the system a single turn around the record gives one picture, so that the videodisc can be viewed in its standard home form as containing either video material, or slides, or some combination of these. Some of the discs can hold an hour's worth of TV, or approximately 100,000 slides.

This very large amount of information comes at a very low cost in volume production. While the cost of making initial masters, a photograph-phonograph process, is considerable, the cost above that for producing an individual record is small. It is estimated that quantity production would produce videodiscs at under $1 per videodisc, counting only the production process itself and not the development of materials.

The videodisc provides a natural environment for holding the multimedia material to use in computer-based learning combinations. Some years ago I called the combination of the computer plus the videodisc player the intelligent videodisc system. Here I am thinking not of the small microprocessor often built into the videodisc player in order to handle the tracking problems, but rather a powerful, stand-alone, personal computer capable of handling all the best present day, computer-aided, learning materials. It would not be difficult to put such a system together today. Several such systems are employed in military projects, such as at Hughes Aircraft Company, Fullerton. The National Science Foundation has recently solicited proposals for developing such systems and preparing teaching materials on them, to gain experience in their use with actual students.

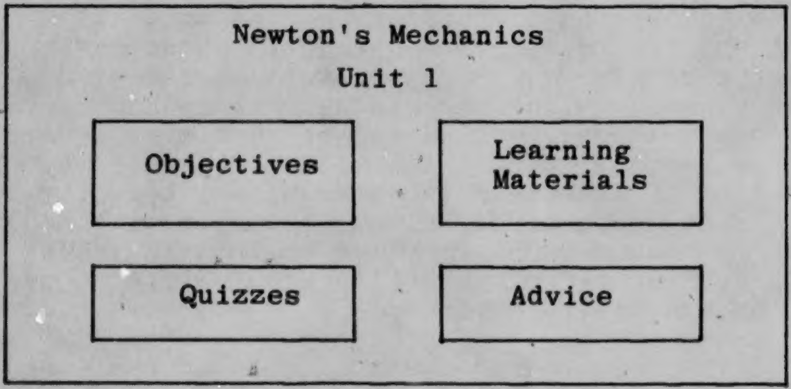
Instruction via the Intelligent Videodisc

Now let us consider what learning material on an intelligent videodisc system might look like from a student point of view. Consider a module in which students are learning about Newton's approach to the laws of motion. This is not a standard approach for typical beginning courses, but one that does bring considerable insight into the process.

By a module I mean approximately one week's work, for the average student, in a mastery-based or personalized system of instruction environment. The notion is that the student will be provided with a clear set of objectives for the unit, with a collection of learning materials, with a guide to these learning materials, and with a repeatable, variable, on-line test to show whether the student has learned the necessary material or still needs to pursue it further. We do not assume that all the material is available on the intelligent videodisc, although much of it could be.

When students first enter the unit they receive a "review," a set of detailed objectives. The objectives will also be available at later times to the students just by asking for them, and also the students will be able to obtain a more detailed explanation of each one, if desired.

The student will move around basically through a map which might look something like this:



You can see that the objectives can be reached from the map.

Another possibility is a discussion of the various types of learning resources available. There will be a fundamental resource track that the student is advised to take, but with many options available to suit the preferences of individual students. Thus, the learning environment will be something of a supermarket environment, not providing a single set of learning materials but providing a rich collection that introduces and reviews the material in a variety of ways.

The objectives and the descriptions of the learning materials will be, since they are not interactive, primarily on artist developed slides. It goes without saying that the full color capabilities of intelligent videodisc systems will be employed, so that the learning material will be as effective as possible.

Within this collection will be modules that develop the material in the usual pedagogical form. These will be similar to existing computer-based learning materials, with the student always taking an active part in the learning process. The general strategy will be similar to that in the Socratic dialogs.

Within such sequences occasionally chunks of film will be shown. In this case an existing film, Newton's Equal Areas, will be broken up into smaller segments. An introductory section in this film introduces the laws of motion in an intuitive form with examples in preparation for Newton's proof of the law of equal areas. After this intuitive sequence is shown on an intelligent videodisc system, a questioning session will check to see whether the student has picked up the critical points about the behavior of particles under the first and second laws. If not, the sequence may be replayed or an alternate computer-based, learning strategy may be encountered. Thus, the film will be viewable in a much more interactive environment than is usually possible; the student could also ask to see the film through in its entirety.

One of the resources to be available on the videodisc will be reading material, the usual kinds of note or text materials. Here we can provide the student with capabilities not available in printed material. Thus, we can let the student pick the line length of the text, and we can let the student have control over delays. In future systems it may also be possible to have the student controlling type fonts, but at the present time we typically have only relatively crude computer-based type fonts to work with. A later possibility will be to allow students to keep their own notes along with the text and perhaps even to read the notes of other students, if the others are willing to have this happen.

As already mentioned, interactive tests are an important component of any self-learning environment. These may occasionally use some of the multimedia capabilities too, but will be primarily descendants of the best examples of current work in interactive, on-line testing. Such tests have tremendous pedagogical advantages in that they can immediately provide feedback to the student and can often provide very detailed assistance when the student is in trouble, based on precisely the type of difficulty the student is encountering. In our work at Irvine we consider the development of effective on-line testing to be one of the most exciting and effective approaches to computer-based learning material, and we expect it to play an equally critical role in the intelligent videodisc era.

The student having difficulty in on-line tests will be given specific aid not only within the test itself but also will be recommended to other learning material. The tests will be related to the competencies or objectives of the unit, and difficulties the student is having will be spelled out in terms of those objectives in an unambiguous form.

We are not by any means exhausting the content of the module. But perhaps enough has been described here to give some view of the nature of such materials that could be made available in the intelligent videodisc environment.

VIDEODISC OVERVIEW

J. D. Fletcher

Defense Advanced Research Projects Agency

This paper presents a very brief overview of current developments in videodisc technology and some guesses about the implications of this technology for education and training. Guesses about these implications are necessarily uneducated because we have had so little "hands-on" empirical experience with videodisc technology in instructional settings. On the other hand, these guesses are worth making because of the immense impact videodiscs may have on instruction. It seems reasonable to discuss first the functional characteristics of the hardware involved since they limit in a very basic way what the contributions of videodiscs can be.

Overview

Bits and pieces of videodisc technology have been with us for some time. In 1966, an IBM 1500 Instructional System was installed in a public school setting for use by investigators at Stanford University (Suppes, 1966). As many ADCIS members recall, this system offered random access to 1,024 16 millimeter still photographs, about 30 minutes of recorded audio information that could be divided into addressable segments of arbitrary length, and a small number of digitally stored graphic characters. The photographic and audio information was stored on tape and the speed with which random access could be accomplished left something to be desired, especially by the kindergarten through second grade students who initially used the system. In the 1968-69 school year, Stanford began using a digitized audio capability in its PDP-10 based beginning reading program (Atkinson and Fletcher, 1972). Speed of random access to the audio messages was never a problem on this system, but storage was expensive. It took 36,000 bits to store one second of audio information. Graphic, photographic, and/or video information was out of the question because the student terminals for this system were teletypewriters--although valient efforts by Paul Lorton (1973) to present graphics on Model 33 teletypewriters for the Stanford spelling program should not go unmentioned. In 1963 the PLATO project had developed a video scanner for its nursing curriculum that provided rapid (less than one second) random access to 128 still photographs. In order to reduce costs, the project developed a capability for rapid random access to 256 still photographs stored on microfiche (Stifle, 1971). This capability was available in 1972 for the PLATO veterinary medicine and elementary reading curriculums. About this same time (1972-73) the BYU/MITRE TICCIT project pioneered the use of computer-controlled interactive television, and issues of computer controlled video information began to arise in instructional design (Bunderson, 1974). An example of this latter is the multi-media system built by the Navy Personnel Research and Development Center which includes (along with almost everything else) random access to video information stored on tape and which allows still frame, slow motion, and reverse motion all under program control (Wisher, 1979).

Disc storage of video information has also been available for some time as NFL and NBA referees must occasionally acknowledge. These discs provide digital storage and a read-write capability of relatively low capacity although improvements in magnetic pick-up heads and magnetic oxide technology have increased this capacity by a factor of about four since the discs first appeared. These discs incorporate most of the features we can anticipate for videodiscs--with the quite notable exceptions of low cost and large storage capacity. Slow motion, still frame, and reverse motion are all available, but there seem to be few examples of these discs being used in education or training.

The two types of videodiscs now in the offing are characterized by the market they are targeted for; there are consumer players and there are industrial/educational players. Various potential manufacturers such as Sony, RCA, IBM, Xerox, Telefunken/Decca, Corning Glass, and ARCO/IOMETRICS are waiting in the wings, but only three are proceeding with definite market plans for videodiscs:

(1) Philips/Magnavox/MCA may begin selling a consumer model player for about $600 in December 1978.

(2) MCA/Pioneer may begin selling an industrial/educational player for about $2,500 in June 1979. About 30 of these players have already been sold as prototypes at about $20,000 a copy (but the price has dropped to about $10,000 recently).

(3) Thompson-CSF may begin selling an industrial/educational player for about $4,500 in March 1979 and reduce the price to about $2,500 late in 1979.

These plans are stated cautiously for all the obvious reasons; the field is developing rapidly, and the plans are subject to sudden and unforeseen changes.

The differences between consumer and industrial/educational systems are significant as Table 1 shows. For instructional settings, the industrial/educational players can incorporate microprocessor control for fast random access, pre-programmed branching, and frame selection. These are currently the players of choice for instructional designers. The features of the consumer players lend themselves to instructional application, but the capability of the industrial/educational players may be sufficiently greater that their higher cost is justified. It may be that, with experience, we will discover what the various videodisc features "buy" in terms of instructional achievement and will thereby be able to recommend commercial or industrial/educational players depending on instructional settings. This, of course, is not likely to occur in the near future. There are few, if any, satisfactory studies that demonstrate in quantifiable terms what the separate capabilities of computer based instruction provide in terms of instructional achievement, and there seems little reason to expect the development of videodisc based instruction to be any different. Most probably we will continue to opt for systems with the largest suite of capabilities available in order to maximize the flexibility allowed instruction designers. It should be emphasized that the question is not whether the industrial/educational videodiscs will be more effective for instruction--they doubtless will be--the question is whether and in what settings their increased instructional effectiveness will compensate for their greater cost.

The three videodiscs discussed above are all optical, laser-based systems. They use a 12-inch disc with a spiral track. The track is pitted with oblong depressions or micropits that vary in accordance with the audio or video information they represent. During playback the disc spins at 1,800 revolutions per minute while these micropits modulate a low power laser focused on the track and thereby generate a signal that is processed and

Table 1

Comparison of Consumer With Industrial/Educational Videodiscs

Features	Consumer Players	Industrial/Educational Players
Cost	$550-650	$2500-4500
Still frame	Manual or automatic	Manual or programmed
Frame random access	Manual	Manual or programmed
	Visual identification	Visual or keyboard selection
Frame-by-frame "stepping"	Manual	Manual or programmed
Variable speed motion (forward or reverse)	Manual	Manual or programmed
Two discrete sound channels	Yes	Yes

passed to a standard video monitor (i.e., to the antenna terminals of a TV set). Present technology allows about 30 minutes of video information to be placed on one disc; although 60 minute discs, physically larger and smaller discs, etc. are options at one stage or another of development. One video frame is stored on each track and there are 54,000 tracks per disc. Video, audio, and still photographic information can all be intermingled on these discs, but they effectively provide rapid random access to 30 minutes of video information, 30 minutes of analogue audio information, 54,000 still photographs, well in excess of 400 hours of digitized audio information, 30 minutes of motion picture information, etc. The point to be made is that videodiscs provide rapid random access to a lot of information inexpensively stored. The information is also inexpensive to reproduce. Once a master disc is made current projections are that reproductions can be produced for at most $4.00 a copy. The Encyclopedia Britannica could be placed on a single video disc; in hard copy book form the Encyclopedia costs considerably more than $4.00 to reproduce.

There are at least two important differences between the MCA and the Thompson videodiscs. First, the Thompson disc is "transmissive" and the MCA disc is "reflective". Light from a laser passes through the Thompson disc and is picked up on the other side; the MCA disc reflects light back up so that light is sent from and received on the same side of the disc. Three implications of this difference in technical approach are: (1) the Thompson disc is "floppy", it can be rolled up in magazines and newspapers, and the MCA disc is rigid; (2) by adjusting the angle of the laser, either side of the Thompson disc can be read with no necessity for physically turning the disc over, if the MCA disc is going to use both sides it must be physically turned over--or there must be a light source for each side; and (3) the Thompson disc is more sensitive to dust and dirt than the MCA disc is. The second important difference between the two discs is that the Thompson disc observes European PAL/SECAM television standards and the MCA disc observes American NTSC television standards. The Thompson disc therefore provides more lines per display, better resolution, better color, and more flicker than does the MCA disc. The major instructional implication of this difference seems to be that because of the better resolution with PAL/SECAM standards, programs that display large amounts of text may be better suited to the Thompson disc. On the other hand, if instruction designers want to take advantage of the millions of already purchased television sets in American homes, schools, and industries, they may be well advised to use the MCA disc.

It is notable that since there is no physical contact with an optical disc during playback, the disc should last indefinitely. Both RCA and Telefunken/Decca have developed systems using a mechanical stylus to read the video and audio information on their videodiscs. Discs read in this fashion are expected to last about 100 hours. Telefunken/Decca has been selling their videodiscs for about three years. Their discs can store about 10 minutes of color video information, and they have a useful lifespan of about 100 hours. The diamond stylus used by Telefunken/Decca also lasts for about 100 hours. The disc is not well suited to still frame, random access and variable speed search, and none of these features are offered. The RCA disc was to store 60 minutes of video information using a two-sided disc. The disc was expected to last for about 100 plays, and the sapphire stylus was expected to last around 200 hours. Still frame, random access, and variable speed search were not to be offered as features of the RCA videodiscs. These statements are all in the subjunctive because RCA has shelved all plans to market its disc--despite initial expectation of marketing it in late 1976 or early 1977.

Three second generation developments in videodisc technology appear to be particularly notable for instructional applications. First is the anticipated appearance of direct read after write, or DRAW, technology. Discs using DRAW technology are likely to be somewhat slower (about 30 minutes) and more expensive (about $10) than other videodiscs to reproduce, but they may be better suited for the small volume experimental development that is characteristic of many instructional settings. Second is the development of still frame display with sound. This capability will require reading into a buffer one track of about 10-15 seconds of compressed speech on one revolution of the videodisc, and then displaying a single track or frame of video information on subsequent revolutions of the disc for as long as necessary. The third development will be that of answer-processing. It is anticipated that about 1K bytes of information needed for answer-processing will be storable in second-generation videodiscs. It should be noted that these are developments that will appear on industrial/educational players; it is unlikely that they will become available on consumer players.

A brief comparison of videotape capabilities with those of videodiscs may be in order at this point. After all videotapes represent a rapidly maturing technology, and they can offer many of the features of videodiscs such as random access, variable speed play, and reverse motion. Three points appear to be salient. First, tapes can be edited and current videodiscs connot. This difference is likely to be removed with the appearance of the DRAW technology discussed above. DRAW technology will, of course, provide editable videodiscs quickly and relatively inexpensively. Second, random access to arbitrary points on a videodisc is much faster than similar branching using videotape. Random access on videodiscs is about 2-3 seconds compared with 20-100 seconds on videotape. Moreover, accuracy of random access on videotape often leaves something to be desired. Third, still frames are possible using either videodisc or videotape, but it is exceedingly expensive and cumbersome on videodisc and single stepping through a series of still frames is essentially impossible. In general, videodisc appears to win out over videotape as the technology of choice for instructional settings, and this is likely to become increasingly true with the appearance of DRAW technology. A far more comprehensive and detailed comparison of video-tape and videodisc was prepared by Heuston (1978).

Videodisc should dominate the delivery of audiovisual materials. Original material to be presented on videodisc can be photographic slides, 8mm, 16mm, or 35mm film, LP records, pre-recorded audio tape, 1/2 inch, 3/4 inch, 1 inch, or 2 inch videotape, or microfiche. The principal step in this conversion process is to get the materials assembled and edited onto a 2-inch master video-tape. From this point a master videodisc which is usually made of glass, is prepared. Probably the first few thousand tracks on the disc will be used for instructional programming instructions. Once the master disc is prepared, reproductions are simply stamped out using a galvanic master duplicating process.

Costs of facilities for preparing the 2-inch master videotape, the glass master disc, and the disc reproductions range from $1,000,000 to $2,000,000. It is unlikely that many instruction research centers or even instruction production centers will purchase their own facilities for this process. This point serves to introduce an aspect of instructional authoring that is peculiar to videodisc materials; the authoring set-up must include a facility for mock-up and simulation of the videodisc materials before they are put into the mastering and reproduction process. An appropriate facility for this process based on a magnetic videodisc could probably be assembled for about $25,000. That is not particularly expensive, especially when compared to $2,000,000, but is it an essential expense that ought to be anticipated by those who plan to get into the instructional videodisc business. An excellent report on authoring instructional materials for videodisc has been prepared by Bunderson, Jarvis, and Mendenhall (1979) under the auspices of the Navy Personnel Research and Development Center.

Table 2 presents a summary comparison of the three videodiscs discussed in this section. The reader should be cautioned that some of the information shown is tentative in the extreme. A more accurate and complete treatment of the emerging videodisc technology is being prepared by Bunderson, Mendenhall, and Jarvis (1979).

Issues

As another paper for this conference points out, one (of many) dictionary definitions of "issues" defines them as "pregnant unsettled matters". The intent of this section is to identify a few of these matters and their accompanying implications as they pertain to instructional videodisc technology. This short list of issues is manifestly incomplete; the hope is that it will be of some heuristic value to those who will be concerned with instructional videodisc development over the next 3-5 years. The list follows in no particular order discernable to the author.

1. In one sense, development and use of videodiscs for instruction flies in the face of current advanced computer based instruction research. Rather than concentrate on the abilities of computers to generate information for students as, for instance, do SOPHIE (Brown, Burton, and Bell, 1975), SCHOLAR (Collins, Warnock, Aiello, and Miller, 1975), and WHY (Collins, 1977), development of videodiscs emphasizes the ability of computers to store and retrieve rapidly and accurately large amounts of information. The distinction between "generative" computer based instruction and videodisc based instruction is analogous to the perennial dilemma of programmers who must trade off processing and memory capacities in allocating computer resources to accomplish the work at hand. Generative approaches emphasize computer processing capabilities by attempting to process

Table 2

Summary Comparison of Three Soon-to-Appear Videodisc Systems

	Philips/Magnavox/MCA	MCA/Pioneer	Thompson-CSF
Cost	$550-650	$2,500	$2,500-4,500
Market	Consumer	Industrial/ Educational	Industrial/ Educational
Cost of Master	$1,500-2,500	$1,000	$1,000
Cost of Copies	$4.00[a]	$1.00	$2.20[a,b]
Technology	Optical-reflective (two-sided aluminum disc)	Optical-reflective (single-sided aluminum disc)	Optical-transmissive (two-sided plastic disc)
Standard	NTSC	NTSC	PAL/SECAM

---

[a] Cost includes reproduction of both sides.

[b] Cost includes $1.00 for protective plastic cover.

unanticipated response from students, whereas videodisc approaches provide immense storage for processing anticipated responses. We need to develop instructional system design procedures with generalizable techniques for using these two approaches to complement one another. Resolution of this issue should yield instructional systems with substantially increased levels of power and flexibility.

2. A most important functional characteristic of videodiscs from the standpoint of instructional design is their ability to provide multi-media presentations at relatively low cost. It may be that the most important feature of videodiscs is not their ability to present vast amounts of separate audio, video, photographic, a relative information cheaply but rather their ability to freely mix digital, audio, video, and photographic informaiton in any conceivable combination. The ability to intermingle these types of information should provide instruction designers with powerful presentation capabilities. An early research priority should be to determine more precisely how effective these new capabilities for mixed mode presentations will be for instruction.

3. The capability of videodiscs for storing immense quantities of digital information should be emphasized. These devices are not going to replace magnetic discs and drums because they lack accuracy, but a missing bit or two in a stream of audio or graphic information should be a matter of small consequence. As stated earlier the amount of digitized audio that can be stored on videodiscs is easily in excess of 400 hours. Those of us who have long wrestled with various devices intended to provide rapid, random, and inexpensive access to audio information should be considerably cheered by the appearance of videodiscs. The implications of these devices for reading, language arts, and second language instruction should not be overlooked.

4. In providing computer based instruction to hearing impaired, or "deaf", students investigators (e.g., Fletcher and Suppes, 1975) often express a need for rapid, random, and inexpensive access to large numbers of still photographs, drawings, or even sign language presentations. The deaf world is visually oriented, and the use of teletypewriters and CRTs create real limits for curriculum designers. Substantial instructional progress has been achieved within these limits, but videodisc technology should have a major, if not revolutionary, impact on education for the deaf.

Oddly enough by using videodiscs for storage of large amounts of digitized audio information, a major, if not revolutionary, impact on education for the blind could also result from introduction of this ostensibly visual technology.

5. In any event, it seems reasonable to expect videodiscs to replace most audio and still photograph devices currently in use with computer based instruction systems. One important exception is not necessarily a device, but it is a capability, namely computer generated graphics which are becoming cheaper all the time. In instruction design, choices between videodisc and computer generated graphics capabilities will have to tradeoff the instructional value of photographic and video information versus graphics and animation. Here as elsewhere we need to develop general guidelines for making case by case decisions. In any case, the computer based instruction community should become familiar with the capability, cost, and availability of videodiscs as alternatives to other devices in configuring computer systems for instructional settings.

6. A few words on standardization are probably in order. Statements on the essentiality of standardization are scattered liberally throughout the videodisc literature. There may be a real consensus concerning this need, but there are very few statements specifying what standards are needed for which videodisc features to accomplish what end. Standards are doubtless needed for operating specifications, electronic interfaces, disc formatting, and user interfaces, but with videodisc technology in such a continuing state of change it may be a mistake to "freeze" some aspects of it too soon. With so many voices being raised for standardization in quite unspecific terms, it may be appropriate to counter with a carefully modulated plea for keeping our options open, at least temporarily, lest some valuable features yet to appear be quashed under the heel of standardization.

7. It may be well to iterate a recommendation implied by comments earlier in this paper. We need to know in quantitative terms what the various features of the emerging videodisc technology, with their various costs both in terms of equipment and human effort buy in quantitative terms of instructional achievement. Decisions by instructional designers need to be made concerning these features; reduction of uncertainty in making these decisions and/or explication of the data and

rationale underlying them should serve to substantially advance the state of the art. As painful as it may be to those of us who enjoy tinkering with exotic new technology and features thereof, there are probably many instructional applications in which the consumer players are the cost-effective alternative of choice.

Final Word

As Heuston (1977) has pointed out, the videodisc may be the most revolutionary instructional technology since the book--for the same reason that the book was effective. Videodiscs capture and make inexpensively available to an unlimited audience portions of the instructional process heretofore available only to the few. An extensive "Delphi" survey by Daynes (1976) suggested that videodiscs will prevent the gap between highly educated and undereducated groups and individuals from widening any further and that videodiscs will facilitate the growth of open learning, access to instruction by the physically limited, and expansion of continuing professional education. There is a requirement for instruction designers and particularly members of ADCIS to attend closely to the development of this dynamic, promising technology.

References

Atkinson, R.C. & Fletcher, J.D. Teaching children to read using a computer. *The Reading Teacher*, 1972, *25*, 319-327.

Brown, J.S., Burton, R.R., & Bell, A.G. SOPHIE: A step towards a reactive learning environment. *International Journal of Man-Machine Studies*, 1975, *7*, 675-696.

Bunderson, C.V. The design and production of learner-controlled courseware for the TICCIT system: A progress report. *International Journal of Man-Machine Studies*, 1974, *6*, 479-491.

Bunderson, C.V., Jarvis, S., & Mendenhall, R.W. *A Study of Authoring Alternatives for Training-Oriented Videodiscs*. San Diego, CA: Navy Personnel Research and Development Center, 1979, in press.

Bunderson, C.V., Mendenhall, R.W., & Jarvis, S. *An Evaluation of Videodisc Technology*. Fort Monroe, VA: Army Training and Doctrine Command, 1979, in preparation.

Collins, A. Processes in acquiring knowledge. In R.C. Anderson, R.J. Spiro, & W.E. Montague (Eds.), *Schooling* *and the Acquisition of Knowledge*. Hillsdale, NJ: Erlbaum, 1977.

Collins, A., Warnock, E.H., & Passafiume, J.J. Analysis and synthesis of tutorial dialogues. In G.H. Bower (Ed.), *The Psychology of Learning and Motivation: Advances in Research and Theory* (Vol. 9). New York: Academic Press, 1975.

Daynes, R.R. *Videodisc Technology Use Through 1986: A Delphi Study* (TR 77-11). San Diego, CA: Navy Personnel Research and Development Center, 1976.

Fletcher, J.D., & Suppes, P. The Stanford project on computer-assisted instruction for hearing-impaired students. *Journal of Computer-Based Instruction*, 1976, *3*, 1-12.

Heuston, D.H. *The Promise and Inevitability of the Videodisc in Education*. New York, NY: WICAT, Inc., 1977.

Heuston, D.H. *A Comparison Between the Videotape and Videodisc as Educational Devices*. New York, NY: WICAT, Inc. 1978.

Lorton, P.V. Computer-based instruction in spelling: An investigation of optimal strategies for presenting instructional material. Unpublished doctoral dissertation, Stanford University, 1973.

Stifle, J. *The PLATO IV Architecture* (CERL Report X-20). Urbana-Champaign, IL: University of Illinois, Computer-Based Education Research Laboratory, 1971.

Suppes, P. The uses of computers in education. *Scientific American*, 1966, *215*, 206-220.

Wisher, R. *An on-line multimedia laboratory for instructional research*. San Diego, CA: Navy Personnel Research and Development Center, 1979, in press.

THE IMPACT OF VIDEODISC ON INSTRUCTIONAL SYSTEMS DEVELOPMENT

WILLIAM W. HAWKINS

Although relatively few videodisc players were available for purchase in 1978, the past year has seen quite a variety of videodisc conferences, solicitations, planning studies, and initial efforts at developing sample lesson materials in order to study the authoring requirements for developing interactive videodiscs. The observations in this paper reflect some of those efforts - primarily in the context of Navy training - to assess what impact videodisc-based instructional delivery systems will have on the overall process of instructional design and development.

First of all, the use of the term "videodisc" should be clarified. Our studies have been primarily oriented around the systems we expect to be marketed in 1979 - the MCA and Thomson-CSF educational/industrial players, along with the Magnavox consumer videodisc. Our study of videodisc authoring requirements for the Navy Personnel Research and Development Center (NPRDC), for instance, was aimed primarily at the Thomson-CSF player (although since that time NPRDC has purchased an MCA player, and Thomson-CSF has changed its microprocessor configuration so that our original 6800 microprocessor control program must be revised). Our aim, however, has been to understand the generic capabilities (and limitations) of interactive videodiscs, regardless of which particular technological approach--optical transmissive or reflective, capacitance, holographic, or film-based--is used in a delivery system.

In addition to the optical videodisc players to be marketed in 1979, we should also note systems at various stages of development by a number of manufacturers. These include:

Players allowing storage of 15 - 30 seconds of compressed audio with each video frame ;

Optical discs for storing up to 10 gigabits of digital information;

Pulse-code-modulation digital audio discs, using a variety of technologies;

High resolution discs for document storage, using either 1,000 or 1,200 scan lines;

Desktop videodisc recorders which allow updating (although not erasing) capabilities for videodiscs;

Film-based videodiscs allowing possible duplication cost of 10¢ - 25¢ per disc; and

Video cards.

The desktop recorders are particularly important, since they promise to drastically reduce some of the fairly substantial current costs involved in videodisc production and revision.

The videodisc must be seen not as a single product, but rather as a variety of products--stand-alone instructional delivery systems, video and/or digital mass storage, new tools for video production facilities, etc. We can expect that the next few years will see videodisc capabilities incorporated into many products with differing capabilities and purposes. The key element in this context is not so much videodisc capabilities, as the computing power to which it is interfaced. The economics of computer hardware and software, and the ways in which digital and video capabilities are interlaced in system design, may prove to be more important factors in product development than the relatively straightforward attractions of the videodisc.

Videodisc-based Instructional Delivery Systems

Table 1 lists several alternatives for videodisc-based delivery systems. The educational/industrial models with or without compressed audio will contain enough programming capabilities to allow for simple branching procedures, but as the requirements increase for more student interaction with the lesson and display, it will become necessary to augment the videodisc with computer-generated text and graphics, preferably with the ability to overlay the digital material onto the video frame. We can probably expect to see videodisc interfaces for all levels of

Table 1

Videodisc-based Delivery Systems--Capabilities and Costs

NO REVISIONS WITHOUT ISSUING NEW DISC

1) Consumer Model

Capabilities - Passive audio-visual programs; programmed instruction still frame with no audio; video with two channels audio student interaction via controls on player.
Probably not engineered for constant use over prolonged periods; maintenance/replacement costs correspondingly higher.

Cost: Approximately $600

2) Educational/Industrial Model

Capabilities - Same as consumer plus simple branching strategies available by storing control programs on disc and using 1,000 bytes random-access memory in the player microprocessor; no audio with single frames.

Cost: Approximately $2,500

3) Educational/Industrial Model with Built-in Compressed Audio

Capabilities - Same as E/I model but with medium-quality audio stored on the player with each single frame.

Cost: Approximately $3,500

REVISION CAPABILITY

4) Educational/Industrial Model with Compressed Audio/Simple Microprocessor Controller/Linked to Slide-Tape Unit/Computer Text-Simple Graphic Overlay

Capabilities - Revisions could be handled in two ways:

a) Text-oriented material could be overlaid or generated as a separate frame on the CRT, while graphics and audio could be handled on slide random access audio units linked to the microprocessor. The slide-tape revisions would have to be arranged sequentially, so that the control code would activate a tape segment and be notified when the segment was finished so that control would return to the videodisc.

b) The slide-tape player could also be activated manually by the student following computer-generated instructions on the CRT screen.

Cost: Approximately $5,000 - $7,000

Table 1 (continued)

5) Educational/Industrial Model with Compressed Audio

Color graphics terminals with full video resolution, microprocessor, and NTSC interface for graphics overlay
Floppy disc with 3.5 Megabyte capacity
Touch sensor
65K RAM or more

Capabilities - Revisions could be handled either by graphics frame generated by the microcomputer software or by digitizing graphics arts and storing it on the floppy disc. A 3.5 Megabyte disc could store around 1750 frames, which would equal around 25% of a 7,000 frame instructional program. The touch sensor would allow "hands on" simulation for part-task procedural training.

Cost: Between $13,000 and $15,000

microcomputer-based terminals, from the simplest home computers to the new raster-scan color graphics terminals. With the growth of the home entertainment/personal computer market and the well-known trends for decreasing costs of processor and memory chips, we can expect to see the integrated home computer/television receiver become a standard household item in the not-too-distant future.

Assuming that there will be digital/video hardware products to meet virtually every need and pocketbook, the key question for the development of efficient instructional delivery systems revolves around software, particularly computer graphics systems with enough power to handle a large share of the instructional graphics load. The development of user-oriented graphics software packages may soon become the key element in strategies to decrease the costs of instructional media programs.

Authoring Strategies

Videodisc systems will provide a wealth of new opportunities for the design of more effective instruction, but these opportunities also bring with them the need for new mixes of skills in instructional design and development teams. Those who have dealt primarily with audiovisual media, such as sound-slide programs or instructional television, are now challenged to incorporate various levels of interactive capability into their lesson materials. Those involved in computer-based instruction now must incorporate the techniques and economics of audiovisual media into their systems.

The integration of the tasks of the artist, programmer, and photographer must be facilitated by new documentation forms which incorporate both traditional audiovisual script/storyboard forms and programming documentation. At each stage of the production process, instructional designer, artist, programmer, and, finally photographer, must be able to link their work with other members of the team through the documentation. Figure 1 illustrates the sequencing of tasks

FIGURE 1 a CONVENTIONAL GRAPHICS

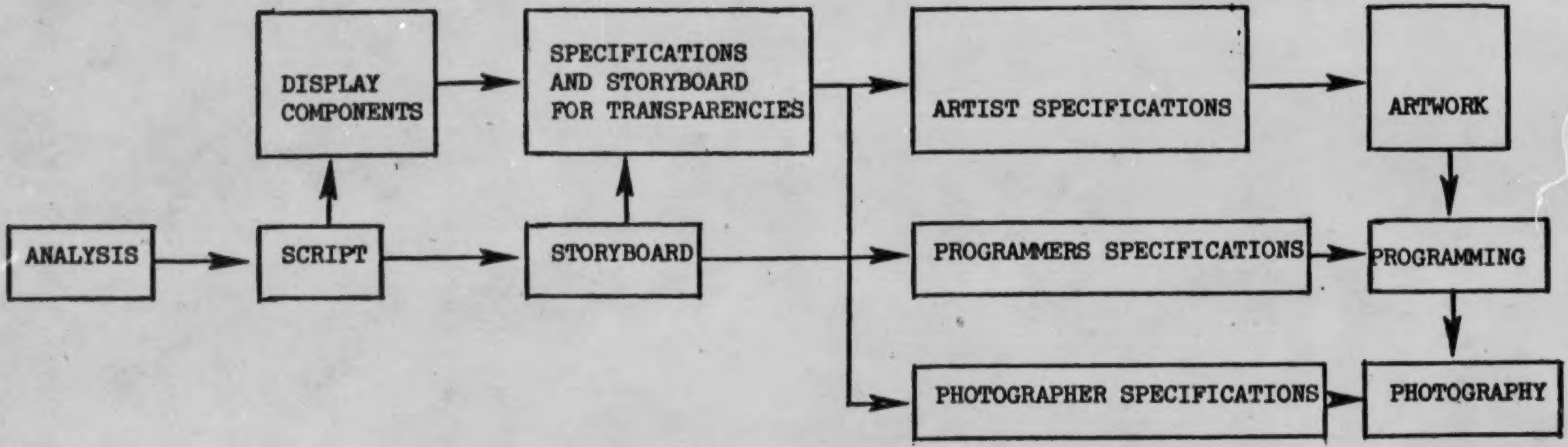

DISPLAY COMPONENTS
SPECIFICATIONS AND STORYBOARD FOR TRANSPARENCIES
ARTIST SPECIFICATIONS
ARTWORK
ANALYSIS
SCRIPT
STORYBOARD
PROGRAMMERS SPECIFICATIONS
PROGRAMMING
PHOTOGRAPHER SPECIFICATIONS
PHOTOGRAPHY

FIGURE 1 b COMPUTER GRAPHICS

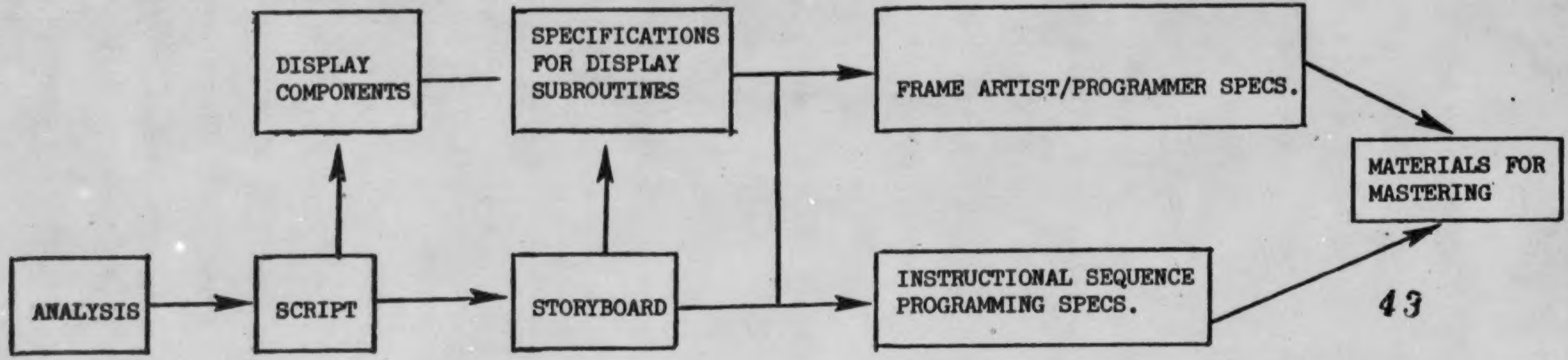

DISPLAY COMPONENTS
SPECIFICATIONS FOR DISPLAY SUBROUTINES
FRAME ARTIST/PROGRAMMER SPECS.
MATERIALS FOR MASTERING
ANALYSIS
SCRIPT
STORYBOARD
INSTRUCTIONAL SEQUENCE PROGRAMMING SPECS.

Figure 1 Graphics Production

indicated by Instructional Science and Development (ISD) experience in converting a segment of PLATO anti-submarine warfare instruction to videodisc format. Whether the artist is working with traditional graphics techniques or with computer graphics software, the tasks of integrating graphics development with instructional design and control program strategies will probably be substantially the same.

The Need for Editing and Revision Facilities

The ability to overlay computer graphics and videodisc graphics and the new capabilities for computer-controlled mixes of still and motion sequences both require extensive tryout and revision before the most effective and economical mix can be determined. At present, for still-frame-oriented material, a 35mm motion picture filmstrip is the medium recommended by the manufacturers for input to the videodisc mastering process. Before the expenses for making the filmstrip and mastering the videodisc are incurred, we must be able to view our material on a CRT; edit, revise, and perhaps test the instructional effectiveness of the lessons with students.

What is needed is a videodisc emulation system that would allow input of a variety of audiovisual media, e.g. slides, videotapes, computer graphics, audio, etc., to be used as still frames or motion sequences under computer control. This is necessary in order to test out quality of audio and visuals, effectiveness of instruction, and to debut the control program. This requires devices such as:

Random-access slide projectors linked to video cameras;
Random-access videotape players;
random-access audio;
computer controller; and
Computer graphics and text overlay.

Until desktop videodisc recorders become available, there are no handy, inexpensive off-the-shelf packages to perform all of these functions. A few such facilities already exist, such as NPRDC's Computer-Controlled Multi-Media System, but most of us will have to engineer such facilities ourselves unless someone develops a product for us.

The revision process: The economics of lesson revision are one of the key factors determining the degree to which videodisc must be augmented by external digital read/write storage. Revision costs, in turns, are determined by mastering/replication fees, the strips or videotapes for input to the mastering process, and, of course, the frequency with which particular instructional material needs revision or updating. Table 2 illustrates a hypothetical 5,000 frame program implemented on a sound-slide system and a videodisc system (with 50 copies required). While these estimates may not be exact, they do indicate the cost-saving potential of videodisc for distribution of audiovisual programs. Costs would be increased, of course, to the extent that the programs are revised to include more interactive capabilities. But what the figures do indicate is that several revised videodiscscould be issued before costs would equal those of the original program using slide duplicates as a distribution medium. If the film-based videodisc technology eventually proves out, it would be possible to issue revised videodiscs for not much more than the present costs of making several slide duplicates.

Table 3 illustrates the break-even point between slide duplication and videodisc for programs of various sizes. Generally, for programs of 1,500 frames and above, videodisc becomes competitive with slide media when as few as 12 copies are required. Even for a potential student audience that is quite small, videodisc will prove to be a more effective distribution medium than slides (or, for that matter, any other existing audiovisual media).

Table 2

5,000 Frame Program--Comparison of Slide and Videodisc Costs (50 Copies Required)

Assumptions: Slide duplication costs - 25¢ per slide
Filmstrip costs - $2.50 per frame
Videodisc mastering - $1250
Videodisc replication - $1/disc (50 discs = $50)

Total Slides Required	250,000	Filmstrip Cost	$12,500
Duplication Costs	$62,500	Mastering	1,250
		Replication	50
		TOTAL	$13,800

Videodisc Replication Costs = 22% of Slide Duplication Costs.

With desktop videodisc recorders, replication costs would approach that of the mastering fees alone, so that videodisc costs would equal approximately 2 - 3% of slide duplication costs.

Table 3

Minimum Number of Copies/Discs Required for Breakeven Point Between Slide Duplication and Videodisc Mastering/Replication

Assumptions: Videodisc mastering/replicating cost - $1300 *
($1250 for master + $1 per copy)

Commercial costs for making filmstrip from slides - $2.50 per frame

Cost of slide duplication - 25¢ per slide

Number of Slide in Program	Number of Discs @ 25¢ per Slide
100	64
250	31
500	22
1,000	17
3,000	12
5,000	13
7,000	11
10,000	11
50,000	12

* These are estimated costs once automated videodisc mastering/replication facilities are operating.

Final Comments

The few examples of videodisc costs presented here are intended to highlight videodisc's attractiveness as a distribution medium compared to traditional audiovisual media. Since our focus until now has been mainly on Navy training programs, where sound-slide tends to be the dominant medium, the impact which videodisc will have on computer-based education costs remains unassessed. From a hardware standpoint, videodisc is simply one more aspect of the microelectronics revolution which is now making inexpensive,highly sophisticated stand-alone instructional delivery systems possible. From the production standpoint, however, the increased graphic capability which videodisc now makes possible from computer-based education will undoubtedly result in an overall increase in production costs for many of us. Graphics artists are expensive, especially outstanding ones. As noted previously, the impact of computer graphics in decreasing graphics production costs will be a key element in determining the economics of production for videodiscs. There are a great many instructional and production strategies which "intelligent" videodisc systems will make possible. Only as we gain experience with this new instructional medium will we be able to accurately assess costs and optimum production strategies. Regardless of costs, however, videodisc-based systems will provide a vastly expanded marketplace in which designers of computer-based instruction can afford to develop highly attractive and effective instructional programs which can potentially reach hundreds of thousands of people.

FIVE PHASES IN THE LIFE OF CBE SITES: CONSIDERATION OF ALTERNATIVES

II. STAFF SELECTION AND RETENTION

Larry Francis
Battelle Memorial Institute

When computer-based education (CBE) centers or projects are described, often the first characteristics listed are the computer hardware, terminals, and software. Next, perhaps, the subject matter areas are used to categorize the activity at a center. Information about the composition of the staff is typically given less attention; nevertheless, investigations have shown that staff characteristics and staff structure have often been the critical features of a CBE center--outweighing hardware, language, etc. in terms of what was potentially and actually achieved.

Roles vs. Skills

In most fields, when considering what staff to hire, an administrator examines the program objectives and begins writing job descriptions for the roles of the staff needed to accomplish the ojectives. In the field of computer-based education, this approach is not always the most practical. Because of the limited availability of people with CBE-related experience, we suggest that staff be selected and described on the basis of skills, rather than by roles(1). Thus, instead of referring to authors, one refers to staff members with programming skills, subject-matter expertise, and/or instructional design skills. Because job candidates are likely to possess a crazy-quilt pattern of skills, premature combining of skills into roles and job descriptions imposes unnecessary constraints on the staff selection task. Skills we have found necessary in most CBE development programs include:(2)

- Administration/Management
- Leadership
- Curriculum coordination

- Subject-matter expertise
- Instructor experience
- Instructional design
- Evaluation/testing
- Programming/CBE expertise
- Editorial skills
- General support

There are several additional advantages for choosing staff by skills rather than by roles. One of the most consistent findings of a study of PLATO courseware development procedures in military and civilian environments was that new courseware development groups changed their organizational structure one or more times during the period examined (3 years).(3) The upset caused by reorganization can be somewhat minimized by emphasizing skills (which change slowly rather than roles (which are changed suddenly by the reorganization). For example, consider a staff member who would traditionally be classified as an author. If he had difficulty mastering the CBE language, the administrator might assign him only subject matter tasks and responsibilities. Thus the staff member would not waste time or feel frustrated by trying to become a well-rounded author, but would use the skill(s) in which he is strong.

Another advantage for using a "skill" orientation is that it may be easier to convince an overworked CBE director who has managerial, leadership, and curriculum coordination functions, that he can continue in a leadership capacity, but delegate other functions to subordinates.

Team Authoring

It has been assumed in the previous discussion that some sort of "team" is being used to develop CBE courseware. Although any multi-person effort might be said to employ a team approach, it is useful to distinguish at least 3 levels of teams.

- Independent authors may divide the content, but do all parts of courseware development themselves.

- One staff member may make the decisions about design, content, and programming, but "subcontract" support staff to do portions of the subtasks.

- No single person may make all the decisions about design, content, and programming. In this situation responsibility for a piece of courseware is shared among the developers. Development is like an assembly line and control passes on when a stage of development is complete.

Advocates of this last form of team authoring often feel that they have found "truth" and "THE method" in comparison to authoring courseware more independently. In fact, the objectives, time constraints, and author backgrounds strongly affect what solution is best for a given situation.

Many courseware development groups start out in the "team" mode described first and gravitate toward approaches with a higher degree of specialization. Often such a pattern is predictable. A loose, independent organization is simple to set up and operate. However, because few staff members possess all the skills needed to be effective authors, uneven, inconsistent courseware begins to emerge. Likewise, staff members' individual strengths and weaknesses are found. In order to enhance productivity and increase uniformity, the manager of courseware development groups then decide to assume the larger administrative burden associated with the second and third forms of the team approach. Thus, the staff's strengths are exploited and the manager achieves greater control.

Selection of the Director

It is predictable that the selection of a group leader or project director has major impact on the success of any endeavor. Because there always seems to be a shortage of qualified managers, there is a tendency to "stretch" the management staff too thinly in order to make maximum use of the available managers. In some cases a manager will try to oversee five or six different programs or projects. Our experience shows that whenever the staff members to be supervised exceeds three fulltime equivalents or five people, a fulltime administrator is generally needed. We have found two other criteria to be useful to determine the appropriateness of the "load" on the administrator:

- The manager should have enough time available to see students using CBE. Unless this time is made available, the manager tends to evaluate the CBE lessons and project only from his perspective rather than as the students/trainees see them.

- The manager should be the appropriate "distance" from the CBE activity: neither too close nor too far. Some managers who have acted as both a CBE developer and also as the manager found it difficult to focus on the long-term administrative issues because the short-term development issues got in their way. At the other extreme, one supervisor found that because of his unfamiliarity with the details of his CBE project, he was left out of the decision-making process. His subordinates found that he could not quickly grasp the issues and hence made the decisions themselves.

Selecting CBE Developers

The CBE developers one can hire typically have one of three backgrounds: subject matter expertise, instructor experience, or a programming background. Each type of person may become a full-fledged CBE author, but each has a potential weakness or difficulty of which to be aware. Subject matter experts may produce lessons which tell, but don't teach. The lessons may be technically sound but lacking interaction, questions, and pedagogy. As a result, the materials are boring. Instructors who become suthors have a better chance of understanding what educational strategies will work with students; unfortunately, robbing the classrooms to develop CBE may be "expensive" in that classroom students must suffer from inexperienced instructors. Former programmers tend to view CBE lessons without a sense of esthetics. If a lesson executes properly, they may be completely satisfied. Students using such a lesson may find displays awkward, introductions missing, and learning difficult. All of these types of people have been successfully trained as CBE developers; the purpose here is to suggest potential areas of weakness which demand the manager's attention or supplementary training.

Fractional Appointments

The subject of fractional appointments can be condensed to one sentence, "In CBE projects to date, there have been too many full-time authors and too few full-time directors". Directors have already been discussed; part-time authors will be treated here. When "drafting" CBE developers from the classrooms for a CBE endeavor, it is often effective to let them retain a fractional commitment to their former position. Such a staff member can perform useful liaison activities: reassuring current instructors, finding student "guinea pigs", squelching rumors, and basically maintaining rapport between the CBE developers and the classroom staff.

One major problem with split appointments is that actual CBE development time tends to be less than that officially scheduled. Because test writing, meetings, grading etc. have firmer deadlines and are more structured activities than CBE tasks, the CBE work suffers. Separate offices and definite time slots for each kind of activity are an aid.

When a large curriculum is being rapidly developed, the need for coordination among developers is so large that staff members working less than half time generally find too little time left over for efficient work. However, small fractions may be fine for staff members not involved in coordination meetings (e.g., subject matter experts or programmers). Part-time student programmers have often been employed with high cost-effectiveness.

Staff Morale and Motivation

Once a staff has been selected and trained, the administrator's next worry is attrition. Loss of trained staff members from a CBE group can be expensive, not only in terms of replacement cost, but also in terms of lost time and unplanned, but necessary, staff reorganizations. Because financial rewards for a superior performance can generally be given but once a year and are subject to many other constraints, smaller non-financial rewards can be used to thank, motivate, and reinforce a deserving staff member. Morale of staff members is enhanced when the following actions are taken:

- Recognition of the contribution of a staff member on the credits page of a CBE lesson or in reports, journal articles, proposals, etc.

- Use of a staff member's CBE lessons during a demonstration. An even greater compliment is to have the staff member give all or a part of a demonstration.

- Releasing resources such as disk space, computer time, and computer sign-on privileges for "side" projects or personal use by staff members or their families.

- Allowing individuals flexibility in their working schedules is possible to some degree in most CBE environments. In fact, because the potential flexibility is obvious to all, failure to offer it may cause discontent.

- An "ego boost" is typically experienced whenever students use a well-tested lesson which an author has written. Students and instructors generally make at least a few complimentary statements and the dwindling number of corrections needed serve to remind the author just how polished the lesson is becoming.

REFERENCES

(1) Francis, L. *Guidelines for Establishing and Managing a Computer-Based Education Site*. Urbana, Ill.: University of Illinois, Computer-based Education Research Laboratory, 1977.

(2) Steinberg, E. R., Avner, R. A., Call-Himwich, E., Francis, L., Himwich, H. A., Klecka, J. A., Misselt, A. L. *Critical Incidents in the Evolution of PLATO Projects*. Urbana, Ill.: University of Illinois, Computer-based Education Research Laboratory, 1977.

(3) Mahler, W. A., Misselt, A. L., Schell, R. M. & Alderman, D. L. *PLATO Courseware Development Procedures*. Educational Testing Service Project Report. Princeton, N.J.: Educational Testing Service, 1976.

FIVE PHASES IN THE LIFE OF CBE SITES: CONSIDERATION OF ALTERNATIVES

Eileen Call-Himwich
University of Connecticut Health Center
Farmington, CT 06032

Lesson Development

An author at a remote PLATO site had just returned home after a one-week course in coding which included a smattering of instructional design. The course was his first exposure not only to coding and instructional design, but to computers of any sort as well. He felt challenged by and enthusiastic about the potential of CBE. He was eager to begin writing his own lessons. He thought writing lessons was tremendously creative and he regarded himself as a creative person.

During the short training course, the instructor had stressed the importance of preplanning as the first step in lesson writing, with particular emphasis given to the careful construction of a list of the tasks the author wanted the student to be able to perform by the lesson's end. During the training course the author had seen the importance of such prior analysis, but once he got home and began writing his first lesson, he found prior planning rather tedious -- a little like having to practice scales when what you'd really like to be able to play is Beethoven's "Pathetique." He tried to decide what material should be included, and how it should be organized but he felt stifled, anxious to really "get down to writing the lesson." He finally announced that he was "too creative" to function this way. His "technique" would be to sit at the terminal and compose, impromptu style, ad.libbing as he went along. This procedure felt much more comfortable. He found himself bursting with ideas. He had seen some fascinating games, and wanted to write a game; he had seen some dazzling displays, and wanted to create some graphics. He struggled, but felt invigorated by the struggle. He labored, but felt challenged by the labor. He rhapsodized, "Writing lessons is so creative and personal -- it's like giving birth!"

Slightly over a year later, he bore a rambling, hopscotching, patchwork of a lesson. Throughout the year most of his time had been eaten away by reworking, reshuffling and rewriting what he'd already done. New material was tacked on to the end of the lesson in a rather "Oh - and-I-forgot-to-mention . . . " manner. He'd constructed a game, but the rules were so complicated, students got too befuddled to play. He had included some graphics, but they were juvenile and cartoonish. Students came to refer to the lesson as "Sesame Street."

That's a useful anecdote (which, by the way, is true) because it encompasses not only a number of common design problems, but a number of managerial problems as well. Before discussing specific design issues, let's look at the ways in which misguided management decisions (and lack of decisions) helped undermine this situation.

This project was staffed by a number of subject-matter specialists representing a wide range of programming and teaching experience. No one was familiar with either the TUTOR language or PLATO capabilities. However, author training time was included in production time. Time and budget being primary issues, authors were sent first to a very short course, and then expected to begin producing curriculum materials immediately. In this way each author was essentially behind from the very beginning. In addition, the time press helped foster the feeling among administrators and authors alike that the only really productive work - or as our author put it "getting down to writing the lesson" - was actual lesson programming. The result was a lack of any standardized development procedure. Each author was left to develop materials in whatever way he saw fit.

Without dampening any "creative" inclination, an alternative approach would have been to adopt a set of project standards to which lessons should conform. Another technique which has proved useful at a number of sites is the development of a driver which could be used as a mainframe for project lessons. Such drivers not only standardize some educational practices (such as use of student objectives and lesson "roadmaps"), but also provide individual authors with a pattern for organizing material and routing students through their lessons. Far from hampering creativity, standardized drivers can free authors time to deal with lesson material at a more innovative level.

Finally, since no one on staff had any knowledge experience with TUTOR or CBE, even novice authors were left with very little programming or design support. Consequently, some lessons remained limited to heavily texted display and student interaction that was easiest both to compose and code.

As for the author in the anecdote, he fell prey to a number of pitfalls that could fairly easily have been averted. His biggest problem was in not knowing how to *think* about developing his lesson. Basically, his thinking centered on the content rather than the student. This orientation is partially responsible for his continuing to tack added material onto the end of the lesson. An alternative (and ultimately time-saving approach would have been to first enumerate the tasks he expected the student to be able to perform and then divide those tasks into their component subtasks. Tedious, perhaps, but in doing so the author provides him/herself with not only an interlocking, hierarchical framework for the lesson material, but also a point of reference by which to weigh the relevance of both content information and possible presentation techniques.

Which brings us to another pitfall. One of this author's problems was the use of various graphics and games for their own sake rather than their educational value or relevance. Without a clear definition of the student goals, the author developed a somewhat short-sighted perspective toward presentation. Graphics and games become an end in themselves.

Common Development Myths

Perhaps this author's greatest oversight was in believing a number of "myths" of lesson development. First, and doubtless most disastrous was the myth that:

An author can improvise a lesson - Our author was well-intentioned, but misguided (and reinforced by administrative attitude) in believing that planning and writing a lesson are somehow separate. That's like saying playing the flute and learning what finger positions produce what notes are two different things.

No one, not even experienced authors, can simply extemporize. While it's true that authors often "compose" at the terminal, they are able to only because they have a firm basis of experience. This author was misled by the outward behavior of more experienced authors. He didn't see the mental organization that had gone on or the fact that they were able to "ad lib" only because, having written a number of lessons, they had practiced and become proficient at the necessary skills. Which brings us to his second misconception:

If you know how to teach, you know CBE.- Any craft is grounded in a certain set of skills/tools at which the craftsman must be proficient. They are the heart of the craft. Just as one would not assume a pianist could automatically play the flugel horn, one should not assume an instructor automatically knows CBE. They're in the same field, but require different skills, approaches, etc. They have different potentials and capabilities and only some similarities.

Presenting is teaching (a lesson in content) - The author also believed that simply by stringing some bits of information together, he was teaching. A more extreme example is the medical CBE site at which the emphasis on content alone resulted in lessons evolving into "page-turners" -- heavily textual, bookish lessons with little student interaction. Students were strongly encouraged to sit at their terminals and take notes. Most did - like modern day scribes, conscientiously transcribing books of their own. Expensive books, to say the least.

Lesson development is a one-shot effort - In the long run, one of the most harmful beliefs an administrator or author can harbor is that lessons are written in a single sweep.

At one fledgling site, the staff (entirely composed of novice authors) disregarded consultants' warnings that lesson development is an iterative process that should be interspersed with numerous trial test runs. The authors chose instead to believe that by having a couple colleagues go through their lessons they could eliminate virtually all programming or design errors.

Within days of unleashing their lessons on students, authors found themselves frantically fixing errors as fast as they were reported so that students could either get out of or get into various portions of lessons. Students were stymied by questions which gave them no clues or help, and no way to move backward or forward. Within months there was a virtually unanimous student mutiny, with nearly all students asking to be reassigned to a non-CBE class.

This example speaks for itself. The time investment in trial test runs can save time in the long run. Authors at this site felt panicked by the fact that these "fix ups" were cutting into what they were made to believe was production time. Debugging was not regarded as "producing." A better approach would have been to schedule trial runs into the project time lines from the very beginning.

Basically, lesson development is a question of managerial and author perspective. From the outset, emphasis should be placed on student-orientation. Production should be seen as a multi-layered, iterative process of planning, programming and testing. Inexperienced authors should be bolstered by coding/design consultants. The model of the crash-course trained, self-sufficient author is as unrealistic as the concept of a one-man expedition to the North Pole.

References

Call-Himwich, E. Myth and Reality: Essential Decisions in Computer-Based Instructional Design. Urbana, Ill., University of Illinois, Computer-Based Education Research Lab., 1977.

Call-Himwich, E. Analysis of Aberdeen Courseware. In A. Himwich (Ed.), Summary and Analysis of Aberdeen CBE Project. Urbana, Ill., University of Illinois, Computer-Based Education Research Lab., 1977.

Call-Himwich, E. An Analysis of POMC Courseware at Sheppard AFB. In Analysis of Sheppard AFB Computer-Based Education Project. A Lynn Misselt and Eileen Call-Himwich. Urbana, Ill., University of Illinois, Computer-Based Education Research Lab., January, 1978.

Francis, L., Goldstein, M., and Call-Himwich, E. Lesson Review. Urbana, Ill., University of Illinois, Computer-Based Education Research Lab., 1975.

Francis, L. *Guidelines for Establishing and Managing a Computer-Based Education Site.* Urbana, Ill., University of Illinois, Computer-Based Education Research Lab., 1977.

Tatsuoka, K., Misselt, A. L., & Maritz, P. Attitudes and Performance of Military Students in Computer-Based Technical Training. In A. L. Misselt (Ed.), *User Attitudes Toward CBE in a Military Technical Training Setting.* Urbana, Ill., University of Illinois, Computer-Based Education Research Lab., 1978.

Alec Himwich
Department of Research in Health Education
University of Connecticut Health Center

Implementation and Site Management

In 1972, the Advanced Research Projects Agency (ARPA) began a large scale investigation of the educational and cost effectiveness of the PLATO computer-based educational (CBE) system. This experiment spanned a period of almost five years and, when fully implemented, placed more than 100 PLATO terminals at the disposal of military users throughout the country. About half of these terminals were used either singly or in small groups to explore limited research questions. The remainder were divided among three military training sites to directly investigate CBE educational and cost effectiveness. These sites shared most of the management and implementation problems of the smaller sites along with special difficulties that accompany the development of substantial packages of CBE courseware. In the smaller sites, for example, the limited research objectives often dictated what instructional strategies, if any, should be employed, whereas the curriculum development sites needed individuals who were capable of resolving instructional design questions within a project's complex environment. Also, with staffs of 14 to 20 individuals at these sites, managers faced the problem of meshing a variety of talents into an effective courseware development team while the smaller sites often involved only two or three individuals. With their larger goals and staff, problems of management and implementation and possible solutions were somewhat more apparent in the large ARPA/PLATO sites than in the smaller ones. It will, therefore, be more informative to focus our attention on the richer experience of curriculum development sites.

The authors of these sites were trained by staff of the Computer-based Education Laboratory (CERL) at the University of Illinois. At CERL, the most striking successes in using the PLATO system had been obtained by individual authors who had possessed whatever subject matter, programming, and instructional design expertise they needed to develop their lessons. Coming from this environment, this authoring model was naturally emphasized. Moveover, in the early 1970's it was popularly believed that one could easily become a competent PLATO programmer. The managers of the large ARPA/PLATO sites initially shared this belief. Consequently, they dispatched subject matter experts to CERL in the happy expectation that in a few weeks time the wedding of their subject matter expertise and PLATO programming competence would produce a vigorous, new generatio of lesson materials for their training programs.

At none of the ARPA/PLATO sites were these expectations realized. At two of the sites, extensive reorganizations giving authors more narrowly defined roles were needed to reverse the trend of slow and inefficient lesson development. At the third site, lesson development times was likewise slow using the self sufficient author model which when compounded with other problems the site experienced led to the abandonment of the initial project objectives.

The unfortunate experience of these sites with this particular model of the CBE author reinforces the dictates of common sense. After all, a self-sufficient author must be highly skilled in several different areas. It is unlikely, therefore, that project managers will find in their staffs an aggregation of individuals who uniformly possess the multifaceted talents needed to be competent self-sufficient authors. In fact, given the diversity in subject matter knowledge, programming skill, and instructional design experience that existed on these projects, one would have least expected the self-sufficient author model to have been adopted. To be sure, part of the blame must be given to the initial training that the ARPA/PLATO authors were given. Nevertheless, the inappropriate use of this model is currently prevalent in projects that did not have an initial exposure to it. It thus appears that some other factor may have also influenced the adoption of this model.

The model of the self-sufficient author is the easiest model for managers adopt for a CBE project staff. Since a manager need only assign entire CBE lessons to authors, he can ignore messy problems of coordination within his staff in the production of each lesson. Thus, it is often adopted in the absence of any thoughtful plan for utilizing project resources. As such it is a kind of default authoring model and is characteristic of projects which lack experienced leadership.

Those ARPA/PLATO project managers who had experience in administration and instructional development did not give their authors a free and independent hand in carrying out lesson development. At best, they adapted lesson development procedures to fit the capabilities of their staff. While they may have favored a particular model of development, the realities of staff selection generally made the use of the favored model impractical. Thus, they were faced with the task of assessing their teams' talents at the outset of their projects and adapting lesson development procedures accordingly.

At two of the ARPA/PLATO sites, after an unsuccessful trial with the self-sufficient author model, the project managers revamped their lesson development procedures to better conform to their staffs.

At one of these sites, there was little expertise with instructional design among the authors. The project manager who had spent a substantial portion of his career as a curriculum designer developed a lesson strategy to be used by all project authors. Thus, the authors who were well versed in their subject matter were freed from design considerations which they were poorly prepared to cope with. Also, since the authors were mostly weak programmers, the manager designated the group's strongest programmer as the "Senior Author" and encouraged the other staff members to refer their difficult programming problems to the Senior Author. This not only freed the authors from another task they were ill-suited to do but it made solutions to the programming problems of readily available the entire authoring staff (Himwich, 1977b).

At the other site, the authors were not subject matter experts but were fairly competent petent PLATO programmers. In this case, the project manager who had long experience with developing instructional materials for military training brought in several experienced instructors to oversee the content and presentation of the lesson materials. In this project, the PLATO programmers were freed from the problems of becoming subject matter experts and instructional designers to concentrate on effective use of the CBE medium. As with the former site, the project manager also placed one team member in the role of programming consultant for the other staff member with the same salutary result obtained mentioned above. (Himwich, 1977a, pp. 45-53). The overall effect of the reorganization was especially impressive. Whereas 650 hours of authoring time were needed for one student contact hour before the reorganization (Himwich, 1977a, p. 40), only 100 hours of authoring time were needed for the same amount of CBE instruction under the revised lesson development scheme (Himwich, 1977a, p. 49).

At the remaining ARPA/PLATO site, the project manager had very little experience as an instructional designer. The only member of the team who had any training in instructional development was not allowed to have a significant impact on lesson development plans (Misselt, 1977, pp.24-25). The project manager intended to treat his staff "as professionals" and thus passed on to them the problem of how to develop their lessons. This approach substantially contributed to the fragmented project effort with slow lesson development times, some highly eccentric and ineffective PLATO lessons, and a disgruntled staff.

While the decision of how to deploy a staff in developing CBE courseware must take into account the various capabilities of the staff members, it must also allow for the constraint of the project completion time. The tighter these constraints the more efficient must be the production methods. Such efficiency is usually obtained by producing prosaic lesson materials which take little advantage of the CBE potential for highly individualized instruction. If time constraints are not great, it is possible to depart from an assembly line approach and assign tasks within the project with an eye to producing CBE materials of high quality.

Whatever the time contraints of a project, it is essential to carefully assess them at the project's outset along with the magnitude of the project. The result of such an assessment should be a timetable for the completion of various project phases. Along with the timetable would be an estimate of about how many authoring hours would be needed to produced one student contact hour of material. Although these timetables and estimates of production rates must have a tentative nature, they at least give a way of measuring progress during the project. Most of the managers of the ARPA/PLATO sites created such timetables and derived an estimate of authoring rates. Some of them erred, however, in not using these estimates to modify their lesson development procedures when their early progress was slower than anticipated. For most projects the completion date is not a variable factor. Thus, when production times are slower than planned, it is only reasonable to modify development procedures rather than pray for an extension of deadlines.

Two examples from the ARPA/PLATO project may emphasize this point. At the first site, lesson production was based on the model of the self-sufficient author. The project manager knew that a production rate of 300 authoring hours per student contact hour was necessary to meet his objectives. When the first lessons took longer than this to produce and when they were judged to be of poor quality by reviewers external to the project, he reorganized his staff to speed up production and improve lesson quality. In this case, the the initial timetable helped to provide a stimulus for reorganizing the staff (Himwich, 1977b). In the other case, an even more complete timetable was drawn up and with it an estimate that 300 authoring hours per student contact hour would be needed to meet the timetable. This estimate had failed to account for non-authoring duties as well as several time-consuming problems that the project suffered in its early stages (Misselt, 1977, pp. 22-25). The initial model employed at this site, actually adopted by default, was that of the self-sufficient author. As time pressures mounted for this project, the manager made no attempt to modify lesson development procedures although such a modification was clearly indicated. (At this site, there was a case of an author who spent over a year working on a single PLATO lesson.)

These two cases not only point up the importance of using one's timetable once it has been derived, but also the damage that authors attempting to be self-sufficient can do to such timetables. Part of this damage originates from the fact that most authors are not skilled enough to be self-sufficient so that they tend to work very slowly as they attempt to improve their skills. However, most of problem arises from the fact that many authors look upon a CBE lesson as an infinitely perfectable work. Thus, since most authors are highly sensitive to shortcomings in their work, they continue to polish them unless their access to their own lessons is actually restricted.

While the most excessive and wasteful expenditures of time and labor occur when a lesson has a single author and when that author has complete automomy in determining when his lesson is complete, a similar sort of problem can arise independent of the lesson production procedures being used. Although it may take on different appearances it always arises from the fact that any CBE lesson can be improved. Thus, a lesson production unit, whether a single individual or a team, can always find ways to continue work on satisfactory lessons. For this reason, it is essential that project managers devise criteria from determining lesson completion and apply it vigorously.

In summary, experience with the ARPA/PLATO project has suggested the following management precepts are very important to the success of any CBE project:

1. The model of author as a self-sufficient combination of programming, instructional design, and subject matter expertise is not viable in most CBE lesson development environments.
2. A division of labor in lesson development should be established to take advantage of available project talents.
3. A list of project milestones and a timetable for their attainment should be drawn up early in the project so that estimates of progress can be made and appropriate remedial actions taken.
4. The determination of when a portionf of CBE coursewae is completed should not reside in the hands of the lesson author or authors.

These ideas may be no more than the dictates of informed common sense. Nevertheless, they were less often followed than not during the ARPA/PLATO project. For this reason and the fact that, by inspection of currently existing CBE development efforts, that ignorance of these ideas is still prevalent today, it can only benefit us to be reminded of them.

References

Himwich, H. A. The Chanute AFB PLATO service test: Site history and management. In H. A. Himwich (Ed.) *Critique and summary of the Chanute AFB CBE project*. Urbana, Ill.: University of Illinois, Computer-based Education Research Laboratory, 1977a.

Himwich, H. A. Some aspects of project establishment and management. In H. A. Himwich (Ed.), *Summary and analysis of the Aberdeen CBE Project*. Urbana, Ill.: University of Illinois, Computer-based Education Research Laboratory, 1977b.

Misselt, A. L. Analysis of Sheppard AFB Computer-based Education Project. In A. L. Misselt (Ed.), *Analysis of Sheppard AFB Computer-based Education Project*. Urbana, Ill.: Unversity of Illinois, Computer-based Education Research Laboratory, 1978.

Five Phases in the Life of CBE Sites
V. Results and Final Evaluations

R. A. Avner
University of Illinois at Urbana-Champaign

The mature Computer Based Education (CBE) production effort constitutes an end point of both a product and a process. By evolution or by simple trial-and-error, mature production groups are most likely to have developed both exemplary applications of the medium and exemplary procedures for efficient production of such applications. Even where these goals are not met, the fact that a group has gone through the full production cycle one or more times means that there may be valuable lessons to be learned from an examination of its history.

The findings reported here are based on longitudinal studies of 22 instructional design groups, cross-sectional studies of 6 additional groups, and observations made during more than 100 evaluations of major CBE design projects. These studies over the past 18 years suggest that it is now possible to make several general statements about the relationships among group productivity, group structure, experience, quality of work, and work environment.

PROCESS CHARACTERISTICS

Mature CBE production groups do show patterns of capabilities and working methods which are distinct from those of beginning groups. In some cases the differences are only cosmetic--involving little more than facility with the jargon and the mythology of the medium. In other cases the differences are more substantial and reveal themselves in highly efficient production of effective instructional material. The most consistent difference seen between beginning and mature groups is, as would be expected, a higher rate of productivity. However this difference is only partially the result of higher levels of experience among individuals within mature groups. Even in groups with relatively high rates of personnel turnover, the established organization can show efficiencies as a result of knowing what is essential and what is not essential in the training of new workers. Established groups also show heavy use of standardized repertoires of procedures which eliminate time consuming exploratory efforts.

Differences between mature groups with equivalent amounts of exposure to CBE seem related to three major factors: (1) the extent and pattern of production demands placed on the group, (2) the variety of instructional settings in which a group produced materials, and (3) the degree of commitment toward producing measurable changes in student performance.

Production demand Studies of individual author productivity (Avner, in press) show that one of the major factors determining production rate is the extent to which authors experience a continued pressure of short-term deadlines for production of material. Authors who have met months or years of 2-3 week deadlines for producing completed CBE instructional modules show production rates that are a several times higher than those of individuals who have never met continual short-term deadlines. Production rates of 30 author hours per hour of student material are not uncommon even in instances where pedagogical structure as well as content must be produced.

Variety of experience The quality of material produced when a group is faced with changes in student population or changes in types of instruction required seems highly related to the range of past experience of a group. Groups which have extensive experience in providing instruction of a single type to a restricted population of students tend to overgeneralize procedures which were found to be effective in that situation. This tendency toward overgeneralization is particularly virulent when the past experience was very successful.

Performance orientation Development of effective instructional approaches by a group seems ultimately to depend on a commitment of the group or its leadership to production of material of demonstrable effectiveness. No group is ever likely to admit a desire to produce poor quality material but many groups demonstrate a head-in-the-sand approach to production which almost assures such an outcome. Groups where frequent, early testing with students is used to gather data which are immediately used in a revision process show a high liklihood of producing both high quality material and a repertoire of procedures for efficient production of similar materials. Groups which concentrate only on measures of quantity of output show a high liklihood of producing both low quality material and a repertoire of procedures for efficient production of similar materials. Unfortunately, quantity is easier to measure than quality and far too many groups gradually yield to use of module or lesson counts as the sole index of group effectiveness.

Administrative considerations also differ in type and importance for new and mature CBE projects. The mature production effort is faced with a class of problems that the beginning group sees in only a limited fashion. Foremost among these problems are those of maintenance, documentation , and updating of existing materials. Some successfully implemented materials have died an agonized death because of insufficient planning for long-range support. With turnover of personnel, inadequate documentation frequently proves the undoing of projects which have to devote more time to upkeep of packages than was invested in the original production.

Confusion between start-up and steady-state costs and the fact that these costs continually change in relative importance during development of a project causes administrative headaches to projects at all stages of life. New groups may either ignore start-up costs or mistake them for steady-state costs. Mature groups are well aware of the burden of steady-state maintenance costs but may mistakenly assume that they have left major start-up costs behind them forever.

PRODUCT CHARACTERISTICS

In a field such as CBE where major learning is still going on among experienced practitioners, groups which communicate effective techniques to other groups can be considered to have produced a product that is at least as valuable as the instructional materials designed for more typical students. In fact, several of the groups studied had research rather than production goals and intended to produce reports which would communicate effective techniques for use of CBE.

Formal reports of instructional development efforts by mature groups were rarely in a form which would convince others of the efficacy of the medium and only slightly more frequently in a form which showed convincing evidence of the effectiveness of the design effort itself. This paucity of evaluative output was particularly evident in academic settings and in projects operating on a job-shop basis. Single-task oriented groups (especially those with research goals) showed a higher incidence of evaluative efforts in final product descriptions; however even these were often limited to provision of evidence that efforts had been expended and products generated.

Attempts to communicate CBE instructional design skills to others by journal articles or technical reports often seemed inversely related to the actual record of productivity of a group. This relationship seems most likely the result of the fact that such reports are often motivated by professional or contractual demands which are rarely present in job-shop production (the setting where the greatest depth and range of experience is likely to exist). In several instances it was found that groups said less and less publicly about methods of CBE design as their experience grew. As a result, the several excellent publications and planning guides available for those seeking initial aid in structuring design efforts are not supported by an equivalent literature giving support for the practical problems of efficient production of effective material.

Both the disappointing lack of formal evidence for effectiveness of materials and the lack of public descriptions of effective CBE applications and techniques are probably a predictable outcome of the implicit goals of most of the projects observed. Production efforts rarely had to validate their materials in a form designed to convince persons other than local consumers. Research efforts often focused on methods designed to work with a wide variety of media, with the result that techniques unique to CBE were sometimes actually avoided. Despite these disappointing trends, effective materials and techniques were developed and were documented in some cases (Avner, 1978).

OUTCOMES

The probability of survival of the mature projects observed seemed to be related more to their reason for establishment than to the quality and quantity of their work. Projects organized for single tasks (e.g. to carry out a contract or produce a course in a given area) showed the highest probability of being dissolved after completion of the task. Given the stated goals of such a project, this outcome is certainly to be expected. Nevertheless, in most cases it turned out to be an outcome unintended by members of the projects. With only rare exceptions, organized groups attempted to extend their existence beyond the completion of the intitial goal by seeking funding for additional projects. Based on the evidence for improved productivity by these mature groups, such efforts may often be in the best interests of the educational community. Unfortunately, where added funding was not immediately forthcoming, experienced members of such groups were usually quickly lured away by other production groups, and the original group was left with insufficient resources to support further production efforts even if the funding became available.

In conclusion, some mature CBE design groups possess skills in kind and quality that go beyond those possessed by beginning groups. These skills have the potential for allowing production of instructional products of superior effectiveness. However, information about the nature of these skills or the effectiveness of resulting products is seldom communicated well.

References

Avner, R. A. Cost-effective applications of CBE, Educational Technology, 1978, 18(4), 24-26.

Avner, R. A. Production of computer-based instructional materials. In O'Neil, H.F. Jr., Issues in Instructional Systems Development. New York: Academic Press, in press.

The Use of MISS in Computer-Based Instruction

W. R. Sanders, A. Levine and R. Laddaga

Inst. for Math. Studies in the Social Sciences
Stanford University
Stanford, California 94306

Abstract

The micro-programmed intoned speech synthesizer (MISS) utilizing linear predictive coding techniques was developed to provide speech output for a major demonstration project of sophisticated computer based instruction (CBI). In addition to simple resynthesis of pre-analyzed recorded speech, the MISS system can concatenate individual words into natural sounding utterances by modifying their pitch, duration and amplitude. Text analysis programs provide the linguistic parameters that direct MISS to apply the appropriate manipulations. MISS has been used with Stanford University CBI courses over the past three years. The MISS system and its CBI uses are briefly described and several evaluation studies are summarized.

1 Introduction

The Microprogrammed Intoned Speech Synthesizer (MISS) system is a collection of hardware and software modules designed and constructed at the Institute for Mathematical Studies in the Social Sciences (IMSSS) to provide access to high quality speech synthesis for users of IMSSS's computer based instruction (CBI) programs. We begin with a brief description of IMSSS's CBI system and how its audio system operates. The CBI uses of MISS are summarized with particular attention given to the implementation of a curriculum author language. Finally, we discuss several evaluation studies.

2 CBI System with MISS Audio

The Institute's CBI system consists of a pair of Digital Equipment Co. PDP-10's accessing half a million 36-bit words of fast core memory and running a version of the TENEX operating system (Myer and Barnaby (1971)). Disks contain the system file storage as well as a direct access area for speech data. Magnetic tapes are used for back-up storage and communication with other installations. There are some sixty display terminals of various types with transmission rates from 1200 to 9600 baud (student terminals operate at 2400 baud). The MISS hardware provides forty-eight audio channels of which sixteen can speak simultaneously.

The MISS system is capable of reproducing speech in two ways: (1) resynthesizing recorded phrases, and, (2) resynthesizing the prosodically adjusted concatenation of individually recorded words. MISS allows for student modification of the speech rate and volume as well as the synchronization of the speech events with display activity. Prosodic adjustment of concatenated words is directed by linguistic text analysis routines and is accomplished by modification of fundamental frequency (F0) contours, duration and amplitude of the words. In conjunction with a program to develop CBI materials (see Section 6), the prosodically concatenated words allow a curriculum author to create and evaluate a new lesson with speech of reasonable quality. Higher quality speech may be obtained by recording the speech required as a series of phrases. Recording phrases is only feasible for curriculum material which is well defined and stable. Should the curriculum be modified, affected phrases would have to be recorded anew by the original speaker. Courses which require the higher quality of recorded phrases lose the benefit of being able to easily modify the curriculum at any time.

3 MISS Implementation

Linear prediction (LP) is a well documented method for the digital representation of speech; for example, see Markel and Gray (1976). A simple description of

LP is that it is a method of modelling the spectrum and voicing of speech. The smooth spectral representation produced by the LP model can be compressed and economically stored for later use. When retrieved, the LP representation can be used to regenerate a high quality version of the original speech.

The guiding consideration in the design of MISS's use of LP was the need for the highest quality sound reproduction possible. Since the principal purpose of MISS is to provide an audio dimension to CBI courseware, it must provide speech which satisfies two related requirements: (1) the speech must allow a student to listen without strain for periods of several hours, and, (2) it must be natural sounding enough to not divert a student's attention from the course material presented. For this goal, we use more precise (and therefore more expensive) digitization and storage of our sounds than commercially available speech synthesis systems. More discussion of quality in MISS is presented in Section 7.

The MISS hardware consists of a pair of very fast microprocessors connected to the main system memory outputting speech to forty-eight terminals. In addition to these microprocessors, an important component of MISS is its library of PDP-10 procedures which allow easy access to speech sounds and their automated processing. The system was designed and built at IMSSS during 1974-1975. Details of MISS implementation can be found in Levine and Sanders (1979).

At present the MISS system is used principally for American English speech and the analysis has been oriented to provide a high quality LP representation for English sounds. We are currently investigating the applicability of this analysis and representation for other languages, e.g., Armenian and Arabic.

4 Prosody and Word Concatenation

MISS is used to form sentences in real time from individual words, by concatenation, with pitch (F0), duration, and amplitude being controlled so as to produce natural sound quality. These modifications of the prosodic specifications of sentences are accomplished with a series of programs that, given a text sentence, determine the syntactic structure of the utterance, interpret explicit semantic markers included in the text, relate the structure and markers to prosodic parameters, and translate the parameters into control functions for real-time MISS word concatenation.

The use of word concatenation was adopted for MISS because we wished to avoid the many problems involved in phoneme synthesis. Some of these problems are:

1) phonetic-segmental interactions (coloring) of adjacent phonemes;
2) duration modification of phonemes in phonetic contexts;
3) change in phoneme quality (e.g., from /ae/ to schwa) due to stress differences in different forms of the word (from "agile" to "agility");
4) difficulty of locating the stress inside a word.

Additional discussion of these issues and solutions to them can be found in Allen (1977) and Klatt (1976). By using word concatenation, we avoid most of the complexities of phoneme synthesis, since each recorded word embodies the interactions and decisions involved in the synthesis as decided by a human speaker.

Word concatenation is itself not completely free from problems.

1) A word may not be in the lexicon. These words must be either spelled or constructed from some source.
2) Word stress may vary in different phrase contexts. Alternative pronunciations of words thus need to be recorded (or constructed).
3) Vowel reduction can accompany low word stress or change in stress placement due to affixing. The segmental nature of a recorded word would change, again requiring an alternative pronunciation to be available.
4) No *a priori* information about the structure of the word is available to the synthesis system. The syllabic structure of words is much more readily available when concatenating phonemes than when a phonetic analysis of each word must be performed to determine such boundaries.

5 Uses of MISS

One of the reasons for incorporating audio messages in a CBI course is that certain sorts of information can most appropriately be presented to the student via an audio channel. For example, it is often useful to make informal comments about formulas or tables which are simultaneously being displayed. In addition, emphasis patterns of speech convey semantic information in a way that is often awkward to reproduce in writing. Finally, it is useful to have a second channel available for communicating with the student and to direct the student's attention.

Two courses using audio are now fully operational. "Introduction to Logic" given three times a year for five units of university credit and taken by some 250 Stanford students each year has been in use since 1976. "Set Theory" is another audio course which has been in operation for the past year. Courses in proof theory and probability theory are nearing completion and an Armenian course is under development. Further discussion of the instructional uses of audio in CBI can be found in Sanders, Benbassat and Smith (1976) and a movie on this topic is available (Sanders and Laddaga (1976)).

6 Author Language

VOCAL (Voice Oriented Curriculum Author Language) (Hinckley et al, (1977)) is designed to facilitate the authoring of curriculums which incorporate highly interactive audio and text presentation. Lessons written in VOCAL are intended to be patterned after the style of informal classroom lectures. VOCAL contains features that allow the author to specify audio messages in several formats acceptable to MISS, and to coordinate the interaction of the audio presentation with material presented visually on the screen of a CRT terminal.

Among the student oriented features of VOCAL are facilities for repeating material just presented, "browsing" through the course (both earlier and later lessons), and changing the speed of speaking. An additional facility, being developed for a course in set theory, is dynamic speech generation from program created text. This facility is used to speak, out loud, summaries of proofs and informal statements of theorems or hypotheses produced by the student as part of the course.

The VOCAL curriculum compiler allows a curriculum author to specify the desired speech simply by including its orthographic representation. For example, when a lesson author wants to have the sentence, "Learning is fun", spoken to the student, he simply includes the line: (S "Learning is fun.") in his lesson file. MISS can produce this speech in two ways: (a) either the phrase (sentence) is recorded, LP analyzed, stored on disk and reproduced on command, or, (b) the phrase is linguistically analyzed and parameters are generated to produce it by concatenating individually recorded words and adjusting the prosodic contour to produce a natural sounding utterance at the time of playback.

7 Evaluations of MISS

An important aspect of the research being done with the MISS system is the evaluation of different aspects of sound perception and acceptance. Once speech synthesis leaves the domain of intelligibility testing, there are many difficulties in evaluating the quality of the synthesis. Objective standards of quality do not exist in this realm. Furthermore, even the subjective ratings that are given can depend on factors that are basically extraneous to the point of the test in question.

One environment that we have utilized for examining the quality of MISS output is the acceptance of CBI with audio by students. In the course "Introduction to Logic", we offer a choice of display-only and display-with-MISS-audio courseware. We have gone to great lengths to make the two versions of the course as similar and compatible as possible. Students are encouraged to try both versions and to switch between versions whenever they find it helpful. They are further informed that their grades in no way depend on which version of the course they used or how often they switched.

We have observed that students prefer audio early in the course and later change their preference to display. A possible explanation lies in the structure of the curriculum, which has a large amount of discursive material early in the course but leans heavily toward drill and practice with derivations in the later lessons. The derivation exercises make little use of audio currently. When a student knows that a derivation exercise is the next part of the course he will select display-only, because the course overhead is slightly lower than for audio, and computer response is somewhat better.

A second explanation is that the audio is slower, since it proceeds at talking speed, than the display-only version. Furthermore, there is no way for a student to interrupt the audio presentation in the middle of an exercise. Students using displays often like to skim or skip parts of the material which they feel confident about. The inability of audio to be skimmed, except by increasing the rate of speaking, makes it less favored than display-only mode. Some students use the audio version of the course for extra help with sections which are particularly difficult.

The quality of concatenated words with prosody is quite good although not as high as that of recorded phrases. Further improvements to the algorithms are a subject of current research at IMSSS.

8 Comparisons with alternative speech systems

Recognition of individual letter sounds is a difficult task for children learning to read because of the absence of context. We designed a test in which three systems of computer-generated speech were compared to each other and a human-voice control. In this experiment the ability of first graders to recognize individual letter sounds as produced by different computer systems was tested. Besides a straightforward statistical comparison of the results obtained, a learning study was conducted, since there is undoubtedly a learning component to the task of recognizing spoken sounds produced by any unfamiliar source.

A second experiment was also performed, to compare the systems on the production of words in order to test a fairly complete list of consonant and consonant cluster sounds, both as initial and final portions of monosyllabic words. The experimental subjects chosen in this case were fifth graders, as the list of words with the desired sounds as components required a wider reading vocabulary than that possessed by most first graders. A full report on this experiment and results is available in Laddaga and Sanders (1977). Only a summary of salient results will be presented below.

Four different systems for computerized speech synthesis were used for these experiments.

1) (MISS) The MISS system, based on recorded and LP analyzed speech, with prosodic modification.
2) (MIT) A sophisticated phonemic synthesizer developed at the Massachusetts Institute of Technology (see Allen (1977) and Klatt (1976)). MIT converts text by rule into the control parameters for the synthesizer and thus into speech.
3) (VOTRAX) (2 systems) The VS6 and the ML1, commercial systems produced by Votrax, a division of the Federal Screw Works, Inc. (see Votrax (1974)).
4) (HUMAN) A human voice recording directly onto high quality audio magnetic recording tape was used as a control.

A sign test was used to examine the rank of the systems in terms of numbers of correct recognitions of letters. The results of the sign tests show that the evidence for the superiority of the human voice over any of the computer systems is significant at the .008 level, as is evidence of the superiority of both MISS and MIT over VOTRAX. Although MISS ranked higher than MIT in this experiment, the sign test comparison between them was statistically inconclusive. Analyzing the test results with a linear learning model, provides estimates of an additional 12 and 54 minutes of exposure respectively for MISS and MIT to meet the criterion of human recorded speech. The same model predicts the need for an additional 11.5 hours of exposure for VOTRAX to meet the criterion. However, the variation in the VOTRAX estimate is such that the prediction ranges from only 3.2 additional hours exposure to a prediction that no learning is occurring.

As in the letter experiment a sign test was used for a pairwise comparison of the systems for the word recognition experiment. The results of the sign tests are that evidence for the superiority of the human voice, MISS, and MIT over Votrax is significant at the .004 level, as is evidence of the superiority of the human voice over MIT. The evidence for the superiority of the human voice over MISS, and of MISS over MIT is significant at the .035 level.

9 Conclusion

The MISS system has proven to be a high quality, flexible and extensible tool for providing computerized audio to CBI programs. LP coding is utilized for storage and synthesis of sounds. Prosodic manipulation of F0, duration and amplitude, directed by linguistic criteria, enables MISS to generate sentences by word concatenation in addition to synthesizing pre-recorded phrases. A continuing evaluation of MISS is being conducted.

Acknowledgments

The authors would like to thank Dr. Patrick Suppes for his guidance and direction. Work on the project described here was carried out under the following grants from the National Science Foundation: EPP-74-15016 A01; SED-74-15016 A02; SED-74-15016 A03; and SED-77-09698, and (research on prosody) under grant G007604941 from the Bureau of Education for the Handicapped.

References

1. Allen, J. 1977, A modular audio response system for computer output, in 1977 IEEE ICASSP Record 77CH1197-3, pp. 597.
2. Hinckley, M., et al, 1977, VOCAL: Voice Oriented Curriculum Author Language, T.R. 291, IMSSS, Stanford, CA.
3. Klatt, D. 1976, Structure of a phonological rule component for a synthesis-by-rule program, in IEEE ASSP 24, pp. 391.
4. Laddaga, R. and W. R. Sanders, 1977, Testing recognition of computer generated speech with elementary school children, in JASA 62, S1 (abstract).
5. Levine, A. and Sanders, W. R. 1979, The MISS speech synthesis system, in 1979 IEEE ICASSP Record (to be published).
6. Markel, J. D. and A. H. Gray, 1976, Linear Prediction of Speech, Berlin, Germany: Springer-Verlag.
7. Myer, T. H. and J. R. Barnaby, 1971, TENEX Executive Language, Manual for Users, Revised 1973 by W. W. Plummer, Bolt Beranek and Newman Inc. Cambridge, MA 02138.
8. Sanders, W. R. and R. Laddaga, 1976, Use of LPC synthesized speech in an instructional application, JASA 60, S1 (abstract).
9. Sanders, W. R., G. V. Benbassat, and R. L. Smith, 1976, Speech synthesis for computer assisted instruction: the MISS system and its applications, in SIGCUE Bulletin, 8.1, pp. 200-211.
10. Votrax Audio Response System VS-6.0 Operators Manual, Federal Screw Works, Troy MI 48084.

INTELLIGENT INSTRUCTIONAL SYSTEMS IN TRAINING

J. D. Fletcher
Defense Advanced Research Projects Agency
Frank Zdybel
Navy Personnel Research and Development Center

Military and industrial training has been receiving increasing attention recently, and this trend is expected to continue for at least three reasons. First, the quality and quantity of entry level manpower have been decreasing substantially. In the last ten years, college entry Scholastic Aptitude Test scores have declined about 5% in mathematical achievement and 11% in verbal achievement. This decline is uniform across the country; it is unrelated to economic, race, or geographic background. Although this downward trend shows signs of leveling off, there is little call for optimism. In an era of increasing technological complexity, the proportion of individuals with high proficiency and basic skills needs to rise, not fall or remain level. Quantity is also severely decreasing. At present about 4,250,000 people are entering the job market each year, by 1990 the number will drop to 3,150,000.

Second, the complexity of jobs has been increasing dramatically in the last 10 years. To take an example from the Army, there is now almost one "system" per person. There is one wheeled vehicle for every four people in the Army, one tracked vehicle for every 20 people, one radio for every six people, etc. Supplying adequate numbers of people to operate and maintain these systems is, needless to say, a challenge for the Army. The growth in complexity is not limited to the Army, of course. Stenographers' pencils and tablets have been replaced by word processing systems which in turn are soon to be replaced by office information systems. Mechanics' intuitions have been replaced by computer based diagnostic systems communicating with computer based ignition systems in automobiles. Fifteen years ago there were 30,000 pages of technical reference materials required to support a fighter aircraft, today 330,000 pages are required.

Third, the costs of training are increasing steadily. The Department of Defense expended over $6.1 billion on formal residential training in FY 1978, of which almost $3.4 billion was spent on post-recruit specialized skill training. These figures do not include on-the-job training, factory training for new systems, "team" training, or field exercises. Outside of increased expenditures for salaries and benefits the single major contributor to these increases is the cost for equipment to be used in "hands-on" training. This is where intelligent instructional systems come in. One major reason these systems are being developed by military and industrial organizations is their ability to simulate in an interactive, veridical fashion equipment that would otherwise have to be purchased and/or used solely for training purposes. Some "hands-on" training will remain necessary, of course, but the amount of it requiring actual, expensive equipment can be greatly reduced. Moreover, because these systems are "intelligent" they offer real promise for attaining increased training achievement with levels of student input that are decreased both in quality and quantity. At this point it may be reasonable to discuss more specifically what intelligent instructional systems are and what we expect to get from them.

Nature of Intelligent Instructional Systems

In discussing intelligent instructional systems it is usually necessary to point out hastily that no distinction is intended between "intelligently" designed instructional systems and "unintelligently" designed instructional systems. Intelligent instructional systems may be as unintelligently designed as any others. Rather the development of these systems is viewed as a specific effort to apply artificial intelligence techniques to CBI in the sense of information structure oriented (ISO) approaches discussed and advocated by Carbonell (e.g. 1970) who contrasted ISO efforts with ad-hoc frame oriented approaches based on techniques of programmed instruction.

By now most of us are familiar with examples of the speed and accuracy with which computers process vast amounts of information. Micro-seconds no longer seem appropriate for describing the speed of many machines; we are down to nano-seconds and the need for pico-seconds appears imminent. The accuracy of computers in accessing billions of bits of information without losing, or "forgetting", one remains impressive, and the amount of information a computer can process seems endless. A PDP-10 can examine 200 million bits in ten seconds; it would take a person over three years to do the same, assuming he could process two bits every second of every day, seven days a week. Still, we like to emphasize that computers are dumb - they only do what we tell them. In a sense, however, and independent of speed, accuracy, and vastness of storage, computers are capable of powerful forms of processing that we often neglect.

Consider the following sentence:

> The man the dog the girl owned bit died.

This is a difficult sentence for people to parse; they are easily lost in its systactic nestings. Yet a computer could very quickly discern, after diving into its recursive routines for processing nested constructions that there was a dog that was owned by a girl, that the dog bit a man, and that the man subsequently died.

Here is another example:

> The man kicked the ball kicked the ball.

This is a perfectly grammatical sentence as any self-respecting machine would discover after reversing an English transformational rule for deleting function words and discerning that a man to whom a ball was kicked, kicked the ball back. In both these examples, a computer is less likely than we are to be confused or distracted, and its ability to process these two examples illustrates real intellectual ability. "artificial intelligence" is after all a poor name for the business of making computers intelligent. Intelligence, or intellectual ability, is really what the field is all about. That theories of intelligence are tested by algorithmization and putting them on computers is merely an issue of methodology, albeit a central one; there is nothing artificial about the capabilities targeted by this work.

These two examples also illustrate the distinction between what John Brown (e.g., Brown, Burton, and Bell, 1974) calls "intelligent" CBI and what might be called "knowledgeable" CBI. We all seem susceptible to the confusion of knowledge and intelligence. We expect someone who is intelligent to know that relativity probably has more to do with physics than with family ties, that battery may involve either an electrical or a legal charge, and that scales are relevant to both fish and music. However, there are many intelligent people who have not heard of the theory of relativity, the legal aspects of fisticuffs, or B-flat minor, just as there are many educated people whose inferential capabilities are less than awesome. A computer program that could process the two examples above would qualify as knowledgeable, and, at least in the second example, demonstrate a clarity of thought superior to most humans.

Consider the following example, taken from Donald Norman (1973):

What was Charles Dickens' telephone number?

A knowledgeable program would search the attributes it had associated with Charles Dickens looking for a telephone number, and/or it would search its list of telephone numbers looking for one associated with Charles Dickens. Presumably no telephone number will be found for Charles Dickens, and this fact will simply be reported. As most human information processors must be aware, there is a more intelligent answer to this query that completely avoids searching lists of telephone numbers and/or attributes. To reply that telephones were not used until after Dickens' lifetime does require some knowledge, but this information could easily be stored by a computer. The real problem is to devise a computer program that knows when this information is relevant.

Another example taken from Abelson and Reich (1969) is the so-called drugstore problem.

You might say:

I went to three drugstores;

A syntactically appropriate response would be:

How did you go to three drugstores?

A semantically appropriate response would be:

What did you buy in the drugstores?

However, the most natural response would be:

What were you trying to buy that the first two drugstores didn't have?

This last response requires an understanding of what Abelson and Reich called the "pragmatics" of language - the penetration beyond literal decoding to plausible implications. The response also illustrates the distinction between a knowledgeable and an intelligent program. The implications of "pragmatics" are broader than Abelson and Reich foresaw; they transcend the problems of natural language processing and encompass all propositional and conceptual systems.

To sum it up, we have the kind of processing capability required by the first two examples and we should use it. We need the "intelligent" processing capability required by the last two examples, and it looks as if we can get it. Advances have been made in the capabilities of computers to process pragmatics. Notably, most of these advances have been made in the context of intelligent instructional systems.

Consider the following student/computer protocol taken from John Brown's SOPHIE system (Brown, Burton, and Bell, 1974). In this example, the student is completing a series of tests on a computer simulated piece of electronic equipment (a regulated power supply) with a fault generated by the computer. Student responses are underlined.

> WHAT IS THE BSE EMITTER VOLTAGE OF THE VOLTAGE LIMITING TRANSITOR?
>
> BASE
> TRANSITOR
> THE BASE EMITTER VOLTAGE OF Q5 IS - .037 MILLIVOLTS.
>
> The spelling corrector has detected and corrected two misspellings. The system must also know that Q5 is the voltage limiting transistor.
>
> WHAT ABOUT FOR THE CURRENT LIMITING TRANSISTOR
>
> THE BASE EMITTER VOLTAGE OF Q6 IS .349 VOLTS.
>
> The meaning of this question depends on the context established by the prior sentence. The system uses this context to determine that the student is interested in not only the voltage measurements but the voltage across a particular junction.
>
> WHAT SHOULD IT BE?
>
> IN A WORKING CIRCUIT THE BASE EMITTER VOLTAGE OF Q6 IS .593 VOLTS.
>
> Again, context is essential in understanding the student's request. Also the program must know what the correct voltage is in a non-faulted system.
>
> REPLACE Q6
>
> I AM GOING TO ASK YOU SOME QUESTIONS ABOUT HOW Q6 IS FAULTED. ARE ANY JUNCTIONS SHORTED?
>
> Before the system replaces a component it asks him what

he thinks is wrong. If he gets at least something right, the system tells him what was actually wrong and goes ahead and replaces it.

I DON'T KNOW

ARE ANY TERMINALS OPEN?

YES

WHICH TERMINALS ARE OPEN?

THE BASE

Q6 IS NOT FAULTED IN THAT MANNER. Q6 NOT REPLACED.

IS IT POSSIBLE THAT Q6 HAS AN OPEN BASE?

The student now has an idea what might be wrong. However, before the system's evaluation procedures can be applied to his idea or hypothesis, the hypothesis must be better specified. To help the student out, the system suggests a plausible specification.

IS THE COMPLETE FAULT MODE EB/EC/BC = OP/OK/OP WHAT YOU INTENDED?

SURE

WHEN THE SETTINGS WERE CC=1.0. CR=HIGH, LOAD=20, VC=1.0, VR=HIGH THAT FAULT CONFLICTS WITH THE FOLLWOING MEASUREMENT. YOU OBSERVED THAT THE OUTPUT VOLTAGE WAS 11.7 VOLTS. IF Q6 HAD FAULT MODE EB/EC/BC = OP/OK/OP IT WOULD BE 23.9 VOLTS. IN A WORKING CIRCUIT IT IS 19.9 VOLTS.

In determining the logical consistency of his hypothesis with respect to the measurements he has already made, a glaring counter-example is discovered.

ALSO:
THAT FAULT CONFLICTS WITH THE FOLLOWING MEASUREMENT. YOU OBSERVED THAT THE BASE EMITTER VOLTAGE OF Q6 WAS .349 VOLTS. IF Q6 HAD FAULT MODE EB/EC/BC = OP/OK/OP IT WOULD BE .712 VOLTS. IN A WORKING CIRCUIT IT IS .593 VOLTS.

The system then points out a more subtle counter-example.

THAT FAULT DOES NOT EXPLAIN ANY OF THE SYMPTOMS YOU OB-

> SERVED BUT IT IS CONSISTENT WITH THE VALUES FOR THE BASE EMITTER VOLTAGE OF Q5 AND THE VOLTAGE AT N1.
>
> Finally the system tells him what is right or at least consistent with his hypothesis.

Clearly, SOPHIE has progressed beyond knowledgeable syntactic and semantic processing of student inquiries and is behaving in an "intelligent" manner. The system tries to create a reactive environment in which the student learns by actively trying out his ideas in interaction with a simulation program. However, the program does more than simply simulate the information to be transmitted, it provides for tutorial feedback and, in effect, for a one-to-one relationship with an "expert" problem solver who helps the student create, experiment with, and debug his own ideas. In a sense the computer "knows" what it is talking about. It makes decisions and interpretations intelligently, well beyond its store of factual knowledge.

Uses for Instruction

The power and pedogogical utility of systems such as Brown's should be self evident. However, as some reviews have indicated, we may be doing well enough with the CBI techniques we have. One might argue that a more appropriate investment strategy for instructional research and development is to continue work on already available techniques rather than attempt to yoke together artifical intelligence techniques and CBI. However, these systems provide intelligent, interactive simulations for use in training, and there are at least four general benefits to be realized from them. Extrapolating from Raser (1969) these are benefits of economy, visibility, reproducibility, and safety.

Economy Intelligent instructional systems should reduce the costs of training equipment, training materials preparation, and individualization. In the case of training equipment, it is obvious that given phenomena may usually be studied more cheaply in a simulation than in its natural setting. For instance we do not need to send a destroyer to sea in order to teach sailors how to operate and maintain its steam plant and propulsion system. Further, we do not need the destroyer, nor for that matter, do we need the steam plant provided we have available a computer based simulation that reacts to the students' decisions in a realistic fashion. In fact, we do not even need a replica simulation of the system -- one that attempts to look as much like the destroyers' steam plant as possible. We can achieve considerable training effectiveness with a symbolic simulation of the system as long as it is both knowledgeable and intelligent.

In the case of instructional materials preparation, it is notable that a difficulty that has inhibited the progress of both programmed instruction and the most common types of computerized instruction is the expense incurred in developing instructional materials for these media. One way to reduce this expense is to automate the production of items, branching strategies, and hints. This automation is possible, but it requires adequate models of the subject matter being taught. These models will become practical and available if current artificial intelligence techniques for representing knowledge can be made sufficiently general to apply to a variety of subject areas. Once these models are available, instructional materials can be generated for presentation either on-line, by computer, or off-line.

Finally, intelligent instructional systems provide systematic, explicit bases for adapting instruction to the capabilities and needs of individuals. Considerable efficiencies in instruction, including the reduction of training time, will be realized by taking account of what the learner knows, concentrating on instructional material that he has not mastered, and emphasizing instructional strategies that are most effective for his cognitive style. If the basis for the instruction is both knowledgeable and intelligent it should be possible either to "map" onto or to model in it what the learner has mastered and what he has not.

Visibility Intelligent instructional systems should increase the visibility of the phenomina being studied in two ways. First, the phenomena can be made physically more accessible. Teaching the operation of electronics equipment is an obvious illustration of this capability. Making the flow of electrons visible is a substantially more realistic goal in a simulated system than in real equipment. Second, intelligent instructional systems may increase the visibility of a phenomenon by separating it from a confusing and chaotic background. Naval tactics provides a good illustration of this capability. If it were necessary to deploy a task force at sea every time there was some tactical training to be accomplished, tactical decision makers in the Navy would learn a great deal about combating seasickness, ascending and descending metal ladders, and finding wardrooms in the dark, but the basic principles of naval tactics would probably remain obscure. These principles need to be taught where they can be separated from the confusion and chaos of fleet exercises.

Reproducibility Intelligent instructional systems allow students to replay chains of events that they could not otherwise observe repeatedly. This replay capability is valuable for at least two reasons. First, it allows a student to build up increasingly accurate subjective assessments of the probabilities governing uncertain outcomes. Subjective assessment of probabilities is particularly important in maintenance troubleshooting. In diagnosing a fault a troubleshooter must balance off the likelihood that a specific fault has occurred against the costs of testing for it. Some tests require major dismantling of the equipment while others require simple diagnostics. Second, a replay capability enables a student to "tinker", i.e. to vary different aspects of the system in ways that yield insights into how it operates. In short, the replay capability permits controlled experimentation by students that is otherwise impossible.

Safety In some ways, intelligent instructional systems permit more realistic training than that available in the physical systems that are the target of the training. One good example of this feature occurs in training aircraft pilots to fly with one engine out of operation. Typically this training is accomplished in lightly loaded aircraft with one engine feathered or idled. Realistically, the aircraft is as likely to experience engine failure with a full load as with a light load, and if an engine fails it will produce a substantial amount of drag which is absent when it is merely idled. The realistic conditions needed for this training are never presented because they are too dangerous, and the result is that pilots are not fully trained to manage an aircraft with an engine failure. Both a full load and the drag of a failed engine can be presented using the type of simulation available in intelligent instructional systems.

Current Developments

Halff (1978) has provided an excellent overview of current and anticipated developments in intelligent instructional systems. Halff categorized these into four areas: intelligent simulation; modeling human instructors; modeling the student; and instruction in information processing.

Work on intelligent simulation ranges from the Generalized Maintenance Training Simulator (GMTS) developed by Rigney, Towne, and King (1974) to SOPHIE (Brown, Burton, and Bell, 1974; Brown, Rubenstein, and Burton, 1976). GMTS interrogates students working with electronics troubleshooting simulations to determine what faults they have identified as likely and what faults they have eliminated from consideration. SOPHIE is considerably more sophisticated. It tries to create a reactive invironment in which a student actively interacts with a tutorial simulation program that helps him formulate, investigate, and "debug" his own ideas using what the designers call an "articulate expert".

Another aspect of current work concerns modeling human instructors. Brown's work incorporates this aspect to some extent, but it is the focus of work by Allan Collins and to a lesser extent of work by Donald Norman. Gentner and Norman (1977) use an advanced "Pandemonium" (Selfridge and Neisser, 1960) model to infer correct and incorrect actions by students and provide tutorial advice in teaching a simple programming language called FLOW. Collins' extensive work has progressed through several versions of SCHOLAR (e.g., Collins, Warnock, and Passafiume, 1975) to WHY (Stevens and Collins, 1977), and has made extensive use of student-tutor protocols to develop tutorial strategies and goals to be incorporated in these systems.

Considerable effort has been expended on modeling the student in computer assisted instruction as reviews by Self (1974) and Fletcher (1975) have demonstrated. These models range from purely diagnostic systems implied by the branching strategies of instruction based on intrinsic programming notions (Crowder, 1959) through the WEST coaching systems (e.g., Brown & Burton, 1977), which evaluate suboptimal decisions by students, and WUSOR (Carr and Goldstein, 1977), which also evaluates decisions by students for optimability but additionally embodies a psychological model that identifies missing cognitive functions, to BIP (Barr, Beard, and Atkinson, 1975), which uses a semantic network to describe relationships between component skills and operations required in BASIC Programming and to generate a uniquely appropriate sequence of problems for each student.

Finally there is the dramatic possibility of improving the basic cognitive functions of students in the sense of learning strategies as discussed in O'Neil's (1978) recent book. As Halff (1978) points out these efforts have a long and unimpressive history, but the advent of intelligent instructional systems offers renewed promise of realizing this possibility. Two relevant systems cited by Halff are those of Papert (1971) and of Rigney, Munro, and Crook (1978). Papert's idea is that teaching people to program computers, in this case using LOGO, may help them develop general abilities for systematic planning and debugging of their own thought processes. Rigney, et al. have been requiring students to explicate their procedures in perusing technical documents to troubleshoot faulty equipment with the idea that these procedures can be debugged and improved in a deep and generalizable fashion.

Development Issues

The dictionary definition of "issues" relevant to this section describes them as "pregnant unsettled matters." The intent of this section is to identify some of these matters and briefly discuss their prospects. In other words, what are the next steps in the development of intelligent instructional systems and why are we not making them? We present five of these.

1. Dealing with imperfectly understood domains. A reason to dislike the recent use of the word "coach" is that it implies ad hoc instruction that can deliver using even the frailest of heuristics. In fact, all of the work done so far has been in instructional domains with very solid theoretical underpinnings. For WUSOR, which aids performance in WUMPUS, various measures of performance can be devined quite easily, and separation of the issues behind good performance is not difficult. The same is true of SOPHIE. What happens when the system is forced to be issue oriented in dealing with unsettled matters? Will we eventually be able to handle large, messy, and poorly understood domains, such as medical diagnosis, legal consultation, or military tactics?

Resulution of this issue requires advances in our current technology of knowledge representation. There are some exciting prospects in view, but a theoretical framework for these problems is just beginning to emerge. Beyond that, there is the brute question of the size of the required knowledge representations. Their scale and complexity will necessitate a whole new order of monitoring and debugging tools for their construction. On the hardware side, machines with enormous memories will be required. With even the largest of the symbolic processors now becoming available, domain size and isolability will continue as important determinants of what can be approached. Also, as representational media become more powerful, they will tend to become less compact. Finally, there are problems of characterizing fragmentary knowledge -- can we successfully represent knowledge with "holes" in it?

2. Simulation for the sake of understanding. In the recent work with "coaching" systems there has been a tendency to concentrate on augmenting instructional simulation aimed at skill acquisition. A valid practical reason for this emphasis is that it is much easier to pick up cues from a student's exercise than to deal with a dialogue. However, this approach tends to finesse the real problems of characterizing understanding. It would be nice to see wome work on extending augmented simulation in the direction of knowledge acquisition. Can we pick up enough cues from a persons's play with a model

to assess his current understanding? How much structuring is needed to keep the interaction productive? Can a student's current state of understanding be characterized as a model? How far could such a system get pedagogically simply by offering up selected activations of a model? The majority of these efforts have confined their use of simulation to animating a task or game environment. The direction for progress is to expand the use of simulation to include exploration and explanation.

The main reason for dependence on simple task or game simulation is the deficiency of current natural language understanding techniques relative to the demands of instructional dialogues. The student must interact with the system somehow, but not primarily linguistically. The context of a task or game provides the alternative channel. The use of simulation for learning by exploration, however, tends to require rather than relieve the need for a natural language interface. The system needs to be able to talk to the student about his exploration. To use simulation as an explanation technique will also require improved dialogue capabilities, not so much for communicating explanation as describing sufficiently well its goals. Another problem involved in approaching this issue, that was touched on lightly already, is our understanding of the role models plan in understanding. Can a person's explorations in a simulation be interpreted into a model that reflects his current understanding?

3. Developing a theory of explanation. We have only begun to investigate the synthesis of explanations, but this capability has been and will remain a key justification for intelligent instructional systems. Goldstein's work (e.g., 1979) leads the way toward a theory of explanation, but we need to make some assults on varied and substantive subject domains if we expect to see further development. The view of explanation synthesis as a planning activity relative to a model of the student is a tempting one.

Of course, the fundamental framework for this development must be our model of understanding. We must have an adequate idea of what occurs when an explanation is understood. With such notions at a sufficient level of detail, it would become possible to think of explanation synthesis in terms analogous to theorem proving and planning. Again, much depends on the level of development of natural language understanding if we expect explanation to proceed as a dialogue. Finally, if we are to be at all effective at explanation, we must be capable of representing a wide range of knowledge about particular individuals, and of constructing such models on the fly. For example, it would be important to know what sorts of analogies specific students are sensitive to. Most attempts at student models have been based purely on characterization of the subject domain. As we attempt to construct more effective explanations, it will become increasingly important to model both individual and general learning processes explicity.

4. Interface of the instructor with intelligent instructional systems. If the most realistic view is that intelligent instructional systems must rely on the instructor to handle the more complicated issues, just how do system and instructor cooperate? If the instructor is responsible for curriculum maintenance, how does he do this when the medium is intelligent instructional systems? Finally, how do the system and the human instructor share their insights about the student?

The main problem in resolving this issue is that intelligent instructional systems have not yet received any extensive use in real educational environments. In some applications, a desirable cooperation feature would be a capability for summarizing completed interactions with the student, either with a domain model, or in terms of change in student models. The problem of curriculum maintenance is more difficult. Training, because of the job specific nature of its objectives, is likely to have heavy requirements in this ares. Ultimately, much will depend on the development of natural language understanding techniques. More immediately, attempts should be made to provide representation editing capabilities that assume some intermediate level of sophistication on the part of the user. Similarly, stopgap attempts must be made to provide an interface between the human instructor and the model of the student maintained by the intelligent instructional system. For the sake of credibility, and to illuminate the problems involved, intelligent instructional systems implementors must cease to pretend that their systems will operate apart from human institutions.

5. Building production intelligent instructional systems. This issue concerns the investment required to build intelligent instructional systems. Not only are the services of expert implementors required, but a substantial cost is also paid in terms of domain expert consultation. If these systems are to become a practical reality, these requirements must somehow be relaxed.

Again, the ultimate approach to this problem lies in natural language understanding. That option for building systems, however, is a long way off. For now, it is more practical to think in terms of providing better tools, such as knowledge representation languages (e.g., Bobrow and Winograd, 1976) and building blocks. One aim should be to facilitate the system building task as much as possible. Some promising work has been completed by Randy Davis'TEIRESIAS system (e.g., Davis, 1976) which interogates a subject matter expert in order to build a production rule representation of his knowledge. More progress toward generative authoring systems is badly needed if we are to provide generalizable capabilities for producing intelligent instructional systems.

Final Word

In summary, intelligent instructional systems can be distinguished from more conventional approaches by the automation of instructional interaction and choice of strategy. The principal instructional capabilities are tutorial simulation and tutorial dialogue. These capabilities require a computer to (1) generate problem statements and solutions, (2) determine efficient sequences, and (3) simulate a variety of situations encountered on the job. The capabilities enable a student to (1) test his own hypothesis concerning the subject matter, (2) probe for information at different levels of difficulty and abstraction, (3) acquire wide experience in minimum time, (4) obtain instructional material generated for his unique abilities and needs, (5) receive instructional aids for partially completed solutions, and (6) receive reviews and critiques of completed problem solutions. These approaches represent real advances in the development of computer based instruction, they are practical, and they are worth pursuing.

REFERENCES

Abelson, R.P., & Reich, C.M. Implicational molecules: A method for extracting meaning from input sentences. In D.E. Walker and L.M. Norton (Eds.) Proceeding of the (First) International Joint Conference on Artificial Intelligence. Bedford, MA: The MITRE Corporation, 1969.

Barr, A., Beard, M., & Atkinson, R.C. A rationale and description of a CAI program to teach the BASIC programming language. Instructional Science, 1975, 4, 1-31.

Bobrow, D.G., & Winograd, T. An overview of KRL, a knowledge representation language. Cognitive Science, 1977, 1, 3-46.

Brown, J.S., & Burton, R.R. Diagnostic models for procedural *bugs in basic mathematical skills* (BBN Rep. 3669). Cambridge, MA: Bolt Beranek and Newman, Inc., 1977.

Brown, J.S., Burton, R.R., & Bell, A.B. *SOPHIE: A sophisticated instructional environment for teaching electronic troubleshooting (An example of AI in CAI)* (BBN Rep. 2790). Cambridge, MA: Bolt Beranek and Newman, Inc., 1974.

Brown, J.S., Rubenstein, R., & Burton, R. *Reactive learning environment for computer-assisted electronics instruction* (BBN Rep. 3314). Cambridge, MA: Bolt Beranek and Newman, Inc., 1976.

Carbonell, J.R. AI in CAI: An artificial intelligence approach to computer-assisted instruction. *IEEE Transactions on Man-Machine Systems*, 1970, *11*, 190-202.

Carr, B., & Goldstein, I.P. *Overlays: A theory of modeling for computer aided instruction*. (Artificial Intelligence Memo 406) Cambridge, MA: Massachusetts Institute of Technology. 1977.

Collins, A., Warnock, E.H., & Passafiume, J.J. *Analysis and syntheses of tutorial dialogues* (BBN Rep. 2789). Cambridge, MA: Bolt Beranek and Newman, Inc., March 1974.

Crowder, N.A. Automated tutoring by means of intrinsic programming. In E. H. Galanter (Ed.), *Automatic Teaching: The State of the Art*. New York: John Wiley & Sons, 1959.

Davis, R. *Applications of meta-knowledge to the construction, maintenance, and use of large knowledge bases.* (Memo HPP-76-7). Stanford, CA: Stanford Computer Science Department, Stanford University, 1976.

Fletcher, J.D. Models of the learner in computer-assisted instruction. *Journal of Computer-Based Instruction*, 1975, *1*, 118-126.

Gentner, D.R., & Norman, D.A. *The FLOW tutor: Schemas for tutoring*. La Jolla, CA: Center for Human Information Processing, University of California at San Diego, 1977.

Goldstein, I.P. The genetic epistemology of rule systems. *International Journal of Man-Machine Studies*, 1979, in press.

Halff, H. *Advanced research in CBI and computer science*. Unpublished manuscript submitted for publication, 1978.

Norman, D.A. Memory, knowledge, and the answering of questions. In R.L. Solso (Ed.) *Contemporary Issues in Cognitive Psychology: The Loyola Symposium*. Washington, D.C.: V.H. Winston and Sons, 1973.

O'Neil, H.F., Jr. (Ed.) *Learning Strategies*. New York: Academic Press, Inc., 1978.

Papert, S.A. *Teaching children thinking* (AI Memo 247). Cambridge, MA: Massachusetts Institute of Technology Artificial Intelligence Laboratory, 1971.

Raser, J.R. *Simulation and Society: An Exploration of Scientific Gaming*. Boston, MA: Allyn and Bacon, Inc., 1969.

Rigney, J.W., Munro, A., & Crook, D.E. *A formative evaluation of a computer-based instructional system for teaching job oriented reading strategies* (Technical Report No. 84). Los Angeles, CA: Behavior Technology Laboragories, University of Southern California, 1978.

Rigney, J.W., Towne, D.M., & King, C.A. *Interactive computer graphics for performance-structure-oriented CAI* (Technical Report No. 73). Los Angeles, CA: Behavior Technology Laboratories, University of Southern California, 1974.

Self, J.A. Student models in computer-aided instruction. *International Journal of Man-Machine Studies*, 1974, *6*, 261-276.

Selfridge, O. & Neisser, U. Pattern recognition by machine. *Scientific American*. 1960, *203*, 60-80.

Stevens, A.L., & Collins, A. *The goal structure of a Socratic tutor* (BBN Rep. 3518). Cambridge, MA: Bolt Beranek and Newman, Inc., 1977.

THE IMPLICATIONS OF A FORMAL MODEL OF THE SYLLABUS FOR A THEORY OF LEARNING[1,2]

Ira P. Goldstein[3]
Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, Mass. 02138

A CAI system based on artificial intelligence techniques is characterized by employing a formal representation of the syllabus as a data structure for driving its tutoring and modelling capabilities. Towards this end, I have been exploring the use of a graph to represent the evolution of procedural skills. The nodes of this graph represent rules comprising a procedural model of performance for the task being tutored, and the links represent evolutionary relationships between these rules. The relationships include generalization, specialization, analogy, and refinement. (Goldstein 1978) considers the guidance provided by the graph for modelling the student and generating explanations. Here my goal is to focus on the framework for learning implied by the graph and its implications for the design of tutoring programs. In particular, I shall explore how the coach can encourage discovery, tutor learning skills, model belief, and estimate learning complexity. This paper is exploratory. Its purpose is to develop guidelines for our implementation of a third generation computer coach for the mathematical game Wumpus.

1. A graph representation of the syllabus has roots in AICAI research.

Scholar (Carbonell 1970), the earliest of the AICAI tutors, employed a graph (semantic net) representation for declarative facts about geography. The graph, however, encoded only domain specific relationships; it did not embody a series of progressively more refined knowledge of geography linked by various evolutionary relations.

SOPHIE-1 (Brown 1973), the next major AICAI milestone, was an expert-based system for the more complex domain of electronic troubleshooting. SOPHIE-1 compared a student's troubleshooting hypotheses for an electronic circuit with that of its embedded expert and offered advice when the student's analysis went astray. It employed a procedural rather than network representation for its electronics knowledge, but this representation was largely a black box. SOPHIE-1 did not have access to a detailed, modular, human-oriented representation of troubleshooting skills. Nor did it have a representation for the genesis of these skills.

---

[1] This research was supported under NSF grant SED77-19279.

[2] This paper has evolved from many fruitful conversations with my colleagues at the MIT AI Lab and with members of John Seely Brown's ICAI group at Bolt, Beranek and Newman.

[3] Now at the Xerox Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, Cal. 94304.

SOPHIE-2, now under development, will incorporate a modular, anthropomorphic representation for the expert's knowledge (De Kleer 1976). This structured expertise serves as a better foundation for expert-based tutoring, but still is not a model of how the student evolved to that level of competence.

BUGGY (Brown 1977), a program for building procedural models of a student's arithmetic skills, does incorporate both a graph representation for the basic skills and some evolutionary relationships. The basic skill representation is a graph with links representing the skill/subskill relationships. The evolutionary component consists of "deviation" links to "buggy" versions of the various skills.

BIP-II (Wescourt 1977), a tutor for programming skills, again employs a network for the basic skill representation, but embodies a different set of evolutionary relationships. There are links for representing analogy, generalization, specialization, prerequisite, and relative difficulty relations. The BIP-II skill netowrk, however, does not include deviation links nor define an operational expert for the programming domain. Rather it employs author-supplied exercises attached to the relevant skills in the network.

MALT (Koffman 1975), a tutor for machine language programming, does include an "expert" for problems composed from a limited set of skills and solved in a tutor-prescribed order. However, MALT's syllabus of skills are related only by the probability with which MALT includes them in a system-generated problem, and not by any evolutionary links. Hence, MALT does not have BIP's ability to choose a problem based on specific relationships to the student's current knowledge state.

The graph representation I employ, termed the *Genetic Graph* (GG), is a descendant of these network representations. Its nodes are the procedural skills of players of varying proficiency and its links include the analogy, specialization, generalization and prerequisite relations of BIP-II and the deviation relationships of BUGGY.

> A potential confusion in terminology may occur here. The term "genetic" is often equated with heredity. However, I use it here in its older sense, namely, the genetic method is the study of the origins and development of a phenomena. This paper is an exercise in *Genetic Epistemology*, the study of the origin and development of knowledge. This enterprise has been articulately advocated by Piaget [Pia70], who considers it the foundation on which psychology should be based.

2. Wumpus serves as our experimental domain.

Designing coaches for the maze exploration game Wumpus (Yob 1975) has proven to be a profitable experimental domain because the game exercises basic skills in logic and probability. The player is initially placed somewhere in a warren of caves with the goal of slaying the Wumpus. The difficulty in finding the beast arises from the existence of dangers in the warren -- bats, pits and the Wumpus itself. Pits and the Wumpus are fatal; bats move the player to a random cave elsewhere in the warren. But the player can infer the probable location of dangers from warnings he receives. The Wumpus can be sensed two caves away, pits and bats one cave away. Victory results from shooting an arrow into the Wumpus's lair; defeat if the arrows are fruitlessly exhausted.

In 1976 we developed WUSOR-I (Stansfield 1976), an Expert-based coach whose advice was based on a set of rules such as these:.

Positive Evidence: A warning implies that a danger is in a neighboring cave.

Elimination: If a cave has a warning and all but one of its neighbors are safe, then the danger is in the remaining cave.

Multiple Evidence: Multiple warnings increase the likelihood that a given cave contains a danger.

Below is an interaction with the WUSOR-I coach which illustrates the fashion in which the coach explains a rule if its employment would result in a better move than the one chosen by the student. The student's responses are indented and underlined.

Hi, Mary. There are 15 caves, 2 bats, 2 pits and 1 Wumpus. You are at cave 15 with neighbors 4, 14 and 0. Brrr! There is a draft. You are near a pit. What now?

> 4

You are at cave 4 with neighbors 15, 14 and 2. Brrr! There is a draft. Squeak! A bat is near. What now?

> 14

(Explanation E1) Mary, it isn't necessary to take such large risks with pits. Multiple evidence is more dangerous than single evidence for pits. Here there is multiple evidence for a pit in 14 and single evidence for a pit in 0. Perhaps we should explore cave 0 instead. Do you want to take back your move?

WUSOR-I was insensitive to the relative difficulty of the various Wumpus skills. In 1977 we took our first steps toward an evolutionary epistemology with WUSOR-II (Carr 1977) wherein the rule set was divided into four subsets or phases. Each phase was concerned with an increasingly more complex, but more powerful class of evidence. The tutor did not describe the rule of a phase until it believed the student was familiar with the rules of the preceeding ones.

Phase 1: Rules for visited, unvisited and fringe caves.

Phase 2: Rules for possibly dangerous, definitely dangerous and safe caves.

Phase 3: Rules for single versus multiply dangerous caves.

Phase 4: Rules for "possibility sets", i.e., keeping track of the sources of dangers.

Phase 5: Rules for numerical evidence.

These phases constituted a coarse genetic epistemology, better than the completely unordered approach of WUSOR-I, but still far from a detailed platform on which to build new knowledge from old in the student's mind.

WUSOR-III, now being implemented, has evolved from WUSOR-II by defining a set of symbolic links between rules that characterize such relationships as analogy, refinement, correction, and generalization. The result is that the "syllabus" of the coach has evolved from an unordered skill set to a *network* of skills linked by their evolutionary relationships.

3. The genetic graph formalizes the syllabus.

Figure 1 illustrates a region of the Wumpus GG. Examine rule SR2. It is a general statement that singly dangerous caves should be preferred to doubly dangerous ones. ER3 is a generalization of particular rules for bats, pits and Wumpi. This is represented in the network by specialization links to SR2B, SR2P and SR2W. The suffix B, P and W indicates a special case rule for a particular danger. (There are inverse generalization links from SR2B, SR2P and SR2W to SR2 which are not shown.) Each of the specializations is connected to its brothers by analogy links. The analogy relationship is defined formally by the existence of a mapping from the variables of one special case rule to another. Finally, SR2 is connected itself a refinement of a simpler strategy that simply preferred safe to dangerous caves. (Goldstein 1978) develops GG theory in greater detail, discussing other evolutionary relationships including induction, deduction, and deviation links.

The GG is playing an important role in our development of computer coaches. The student is modelled in terms of regions of the graph, rather than simply subsets of the expert's skills, thereby allowing the model to represent simplifications, specializations, and deviations of the expert. Tutoring is improved by employing the genetic links to generate a larger number of explanations of a particular skill. Below are three variations on the earlier explanation given to Mary that the graph makes possible. E2 emphasizes a relevant analogy, E3 a generalization, and E4 a relationship to an earlier point of view.

> ***Explanation E2:*** *Mary, it isn't necessary to take such large risks with pits. We have seen that multiple evidence is more dangerous than single evidence* ***for bats.*** *...*

> *Explanation E3: Mary, it isn't necessary to take such large risks with pits. Multiple evidence is more dangerous than single evidence for all dangers. ...*
>
> *Explanation E4: Mary, it isn't necessary to take such large risks with pits. In the past, we have distinguished between safe and dangerous evidence. Now we should distinguish between single and multiple evidence for a danger.*

The graph does not solve the coaching problem. There are many subtleties in modelling and explaining. How can the coach induce a model of the student; when should it interrupt; how much should it say? The graph does, however, imply a certain framework for learning. This is the subject of the remainder of the paper.

4. The genetic graph is a basis for learning.

The model of the student suggested by the genetic graph is shown in figure 2a. The processes of the student are divided into two homunculi -- a problem solving specialist and a learning specialist -- with the graph serves as the student's basic memory structure for procedural knowledge. The problem solving homunculus applies the program defined by the frontier of his genetic graph to the current task. The learning homunculus extends the genetic graph in response to observed difficulties of the current program, new tasks and tutorial advice.

> I use the term "homunculus" to emphasize that the learning and problem solving components are envisioned to be machines of exactly the same power. Their only difference lies in their programs.

The learning homunculus consists of a set of strategies corresponding to the various links of the graph. Its task is to build new rules, leaving behind -- as a record of its operation -- links which connect the new rules to their evolutionary predecessors. The links are labelled with the learning strategy responsible for the construction.

The links are left behind because they themselves can serve as input to the learning strategies. The existence of a profitable analogy can suggest that more analogies "of an analogous kind" are possible. For example, an analogy between the rules of bats and pits can suggest a similar analogy between bats and Wumpi. It may not be exact, but the suggestion offers a direction for the learning homunculus to explore.

It is, of course, a simplification to believe that the entire genetic graph remains available to the learner. In fact, there is, indeed must be, a process of forgetting. This process must exist partly to avoid an indefinitely growing use of space and partly to eliminate outdated knowledge that would serve only to misguide the learning processes. A theory of forgetting is crucial to an overall theory of learning and of teaching, but goes beyond the scope of this paper.

The genetic graph offers only a structure for a learning theory. It suggests that the learning processes consist of procedures which generate the various links, but it does not describe the details of these processes. It does not enumerate what criteria are used to form analogies, recognize deviations, induce generalizations, or construct conceptual refinements. Discovering such criteria has been the focus of much work in AI. See, for example, Evans (1968) on analogy; and Sussman (1975) on debugging.

However, even without describing the details of these processes, the graph is of use, for it focuses our attention on issues involving the interaction of the teaching and learning processes. Four of these issues which I discuss below are: (1) the student as an active agent, (2) a genetic graph for learning, (3) a theory of belief, and (4) the topology of the graph as a measure of learning complexity.

5. The student is an active agent.

The model of the student presented above emphasizes the viewpoint that the student is an active agent, engaged in a constructive process of generating new knowledge. From this perspective, the tutor's objective is to encourage this process in the student. This reminds us that the current activity of most AICAI tutors -- intervening and supplying a complete explanation -- is only one end of spectrum of tutoring activity. At the other end of the spectrum is "tutoring without talking", that is saying nothing at all, but instead altering the problem domain in order to facilitate the learning process.

"Tutoring without talking" is exemplified by one option WUSOR-II can exercise. It can alter the complexity of the Wumpus game by varying the number of dangers, the propagation distances of their warnings, the number of arrows, and the geometric complexity of the warren. WUSOR-II does this in accordance with its estimate of the student's current level of skill.

A range of intermediate interventions also exist between supplying a complete explanation and not intervening at all. An example is that the tutor could suggest that a rule exists that could be applied in the current situation which is analogous to some already acquired rules, but not specify the new rule or stating the analogy.

The next generation of AICAI tutors should be able to supply advice across this spectrum, altering the nature and extent of their intervention in relation to the current state of the student's knowledge.

6. A genetic graph for learning skills is possible.

Dividing the student into a Learning Homunculus and a Problem Solving Homunculus raises the question of whether the skills of the Learning Homunculus can themselves be represented as a genetic graph. If there are a collection of rules that define the processes of analogy, generalization, debugging and refinement which are themselves related by genetic links, then explicating this graph becomes an important goal.

The competing hypothesis is that the learning processes are not related one to another, nor do they have simplifications from which they evolve. They are only an unconstructed collection of heuristics, acquired in an isolated fashion. I believe this unlikely, but it may not be simple to explicate a genetic graph for learning.

Establishing the existence of a L genetic graph suggests that the L homunculus does not in turn require an L^2 homunculus. Rather the links are the same ones as occur in the domain graph. Hence, the L homunculus is potentially able to operate on its own L graph. Thus the recursion of homunculi is terminated. If this is so, it would be an important result both for artificial intelligence and for psychology, namely that a single learning theory is sufficient for both domain knowledge acquisition and a recursive improvement in the system's own learning capacities.

Naturally for the process to begin, there must be some learning strategies that are innate. Establishing from an AI standpoint which strategies are sufficient to generate the remainder then becomes an important research question.

Given a detailed account of the learning processes themselves, the possibility arises that the Coach might be able to tutor these very skills. This will be an important direction for future research, since tutoring the skills of any particular domain is less important than tutoring the processes by which these skills are acquired.

7. A belief measure can be defined on the genetic graph.

Presenting both a Learning Homunculus and a Problem Solving Homunculus focusses our attention on the relation between the two: in particular, it raises the question of when a new rule added to the genetic graph becomes a part of the problem solver's program. It is a simplification to speak of the program of the Problem Solving Homunculus being the frontier of the genetic graph. A new rule may represent a misunderstanding, may not be an improvement, or may be as yet incomplete. Hence, some inertia is desirable in a dynamic learning system, if it is not to oscillate wildly or degrade its performance by accepting premature modifications.

This corresponds, perhaps, to the psychological observation a student does not always employ a skill which has just been explained. While the student may be able to repeat the explanation, and even describe implications of the new knowledge, he may not actually use the skill when solving problems. Teachers recognize this property of students and employ the heuristic of supplying further examples and different kinds of explanations. Authors of script-based CAI systems can incorporate this educational heuristic by supplying multiple exercises and explanations. But the scripts do not provide a theory of where such additional advice will be needed.

A formal representation for this learning conservatism can be added to our learning model by introducing a belief measure. We can restrict a skill on the frontier from being employed by the problem solving homunculus until "belief" in this new piece of knowledge exceeds some threshold, where "belief" is a function of the number, kinds and recency of explanations and examples that have been provided. In terms of our genetic graph representation, we can say that a new rule is not employed until its linkage into the genetic graph is sufficiently strong, i.e., belief in the rule, defined in terms of the number and kinds of links that attach the rule to the existing graph, exceeds some threshold.

> This a first order theory. The linkage strength depends as well on the number of situations in which the rule has been explained, the time since these links were constructed, etc. However, this first order theory is sufficient to define some interesting learning complexity criteria which I discuss in the next section.

A belief metric can improve the tutor's expectations of the student use of a rule following its introduction. The Psychologist module can maintain a record of its estimate of the student's belief in a rule in terms of the types of explanations provided, their recency, and their number. When belief is below a threshold, the tutor can expect that more explanations will be needed and that the student will be able to describe the rule when queried, but probably not employ it. This threshold can be dynamically adjusted on the basis of the student's performance.

Given this model of belief, we can undertake a fine grained analysis of belief criteria in learners. For example, for some students, many examples of a few links may engender stronger belief than single examples of many links. By examination of the student's performance with respect to the occurrence of such bonds, we can explore the tradeoffs between diverse explanations types, repetition and recency. There is also the corresponding AI question of which belief metrics result in a reasonable learning rate, which lead to instability, and which are too conservative.

8. The genetic graph topology provides a learning complexity measure.

Focussing on the genetic graph as a record of the learning process suggests a relationship between various topologies of the graph and learning complexity. The utility of this characterization is that it provides guidance to the Tutor regarding which areas of the syllabus require more attention and to the Psychologist with respect to which skills the student can be expected to have difficulty with.

From a learning viewpoint, the complete genetic graph of the tutor is a roadmap. It describes various paths the student's learning process might take. If the tutor's graph shows that a given rule has many links, then the expectation is that the student will have little difficulty in acquiring that rule himself. There are many opportunities for him to do so. But if another rule has but one link to the other rules, or indeed none, then here is a topology that suggests the need for tutoring advice.

For example, figure 1 showed a cluster of rules densely connected by generalization and analogy links. Our belief metric suggest that such clusters are easier to acquire than sparsely connected regions of the graph. The procedural import of graph density is to cause the Tutor to expect that repetition will be little needed in dense regions but strongly demanded in sparse areas.

Thus, topologies of the syllabus suggest a theory of learning complexity. Experiments are needed to determine if this is borne out. But if so, it is an important theoretical idea for education, independent of the use of computers.

9. Designing computer students is a research methodology.

We intend to explore the many issues raised here by extending an existing "student simulation testbed' to include computer students which learn. The existing testbed is currently implemented to explore the performance of computer students defined in terms of different regions of the GG. These students are static and do not learn.

Dynamic students can be used to explore the effect of different belief metrics on stability and of different learning strategies on the growth of the graph.

Ultimately, embedding a learning capacity in the coach have an important consequence for the genetic graph itself. It can eliminate the requirement that the AICAI tutor have a complete graph to teach. The graph can be incomplete but grown by the embedded learning program when needed (figure 2b).

10. The genetic graph is not a complete theory of learning.

While the issues raised in this section are provocative, the genetic graph is by no means a complete theory of learning. Hard questions remain to be studied: When should a learning strategy be applied? How are profitable analogies, generalizations and refinements detected? What are the criteria for forgetting? Furthermore, there is an enormous amount of experimental exploration that must be done. But I believe it is clear that AICAI programs will gain increased leverage by embodying an explicit theory of the learner.

11. References

Brown, J.S., R. Burton & F. Zdybel, "A Model-Driven Question-Answering System for Mixed-Initiative Computer-Assisted Instruction", IEEE Transactions on Systems, Man, and Cybernetics, May 1973, 248-257.

Brown, J.S., R. Burton & K. Larkin, "Representing and Using Procedural Bugs for Educational Purposes" Proceedings of 1977 Annual Conference, Association for Computing Machinery, Seattle, Oct. 1977, 247-255.

Burton, R. & J.S. Brown, "A Tutoring and Student Modelling Paradigm for Gaming Environments", in R. Coleman and P. Lorton, Jr. (Eds.), ACM SIGCSE Bulletin, Vol. 8, No. 1, Feb. 1976, 236-246.

Carbonell, J., "AI in CAI: An Artificial-Intelligence Approach to Computer-Assisted Instruction", IEEE Transactions on Man-Machine Systems, Vol. MMS-II, No. 4, December 1970.

Carr, B., Wusor II: A Computer Aided Instruction Program With Student Modelling Capabilities, MIT AI Memo 417, May 1977.

De Kleer, Johan, Local Methods for Localizing Faults in Electronic Circuits, MIT AI Memo 394, Nov., 1976.

Evans, T., "A Program for the Solution of Geometric-Analogy Intelligence Test Questions", in M. Minsky (Ed.), Semantic Information Processing, Cambridge, MA: The MIT Press, 1968, 271-353.

Goldstein, I. P., "The Genetic Epistemology of Rule Systems", International Journal of Man Machine Studies. In press.

Koffman, E. & S. Blount, "Artifical Intelligence and Automatic Programming in CAI", Artificial Intelligence, Vol. 6, 1975, 215-234.

Stansfield, J., B. Carr & I. Goldstein, Wumpus Advisor I: A First Implementation of a Program that Tutors Logical and Probabilistic Reasoning Skills, MIT AI Laboratory Memo No. 381, Sept. 1976.

Sussman, G., A Computational Model of Skill Acquisition, New York: American Elsevier, 1975.

Wescourt, K., M. Beard & L. Gould, "Knowledge-Based Adaptive Curriculum Sequencing for CAI: Application of a Network Representation", Proceedings of the Annual Conference, Association for Computing Machinery, Oct. 1977, 234-240.

Winston, P., "Learning Structural Descriptions from Examples", in P. Winston (Ed.), The Psychology of Computer Vision, New York: McGraw-Hill, 1975.

Yob, G., "Hunt the Wumpus", Creative Computing, Sept./Oct., 1975, 51-54.

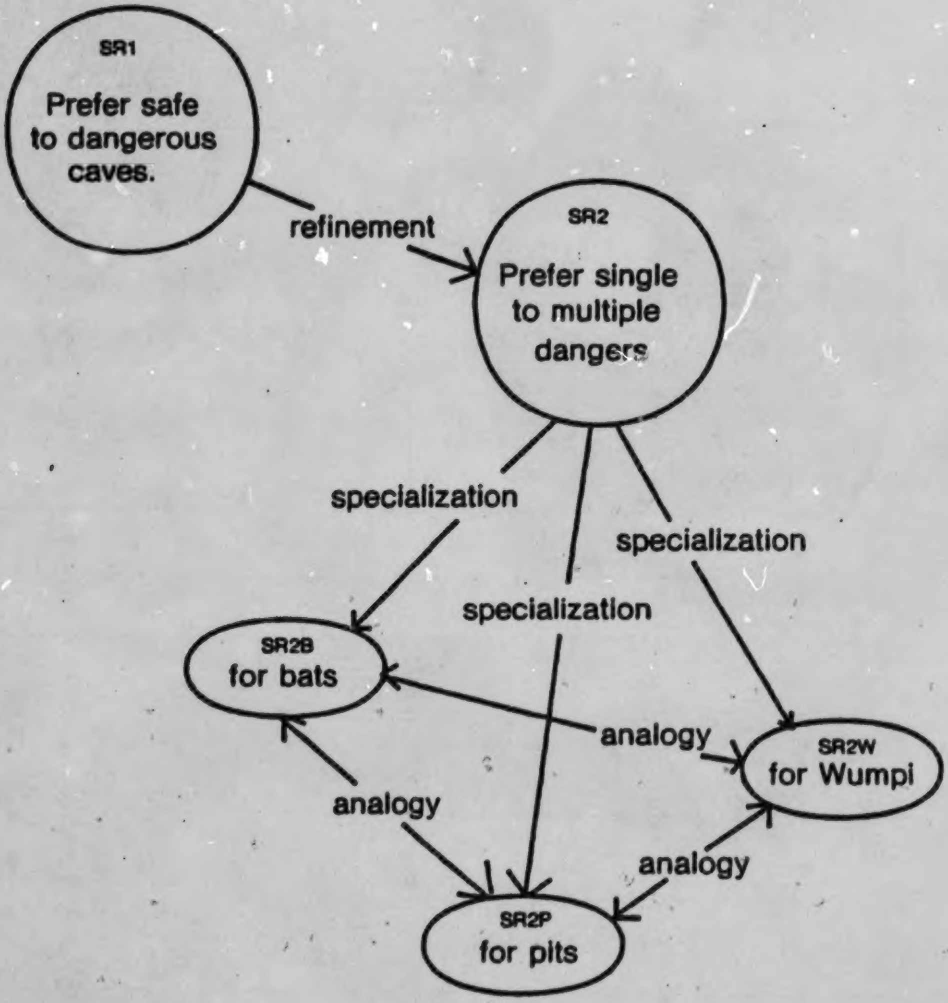
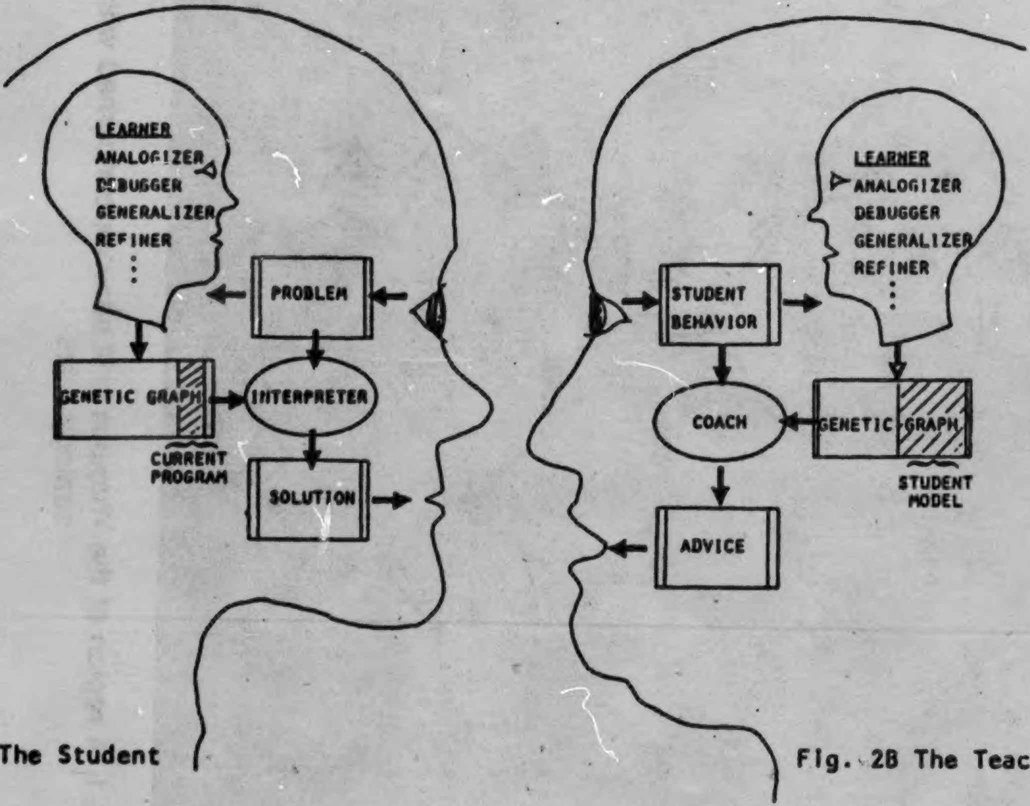

SR1
Prefer safe to dangerous caves.
refinement
SR2
Prefer single to multiple dangers
specialization
specialization
specialization
SR2B
for bats
analogy
SR2W
for Wumpi
analogy
analogy
SR2P
for pits

Fig. 1. A region of the Wumpus skill network concerned with strategy rules.


LEARNER
ANALOGIZER
DEBUGGER
GENERALIZER
REFINER
PROBLEM
GENETIC GRAPH
INTERPRETER
CURRENT PROGRAM
SOLUTION
LEARNER
ANALOGIZER
DEBUGGER
GENERALIZER
REFINER
STUDENT BEHAVIOR
COACH
GENETIC GRAPH
STUDENT MODEL
ADVICE

ig. 2A The Student

Fig. 2B The Teacher

Fig. 2. Homunculus models of student and teacher.

TOWARDS THE INTELLIGENT EVALUATION OF COMPLEX COMPUTER BASED LEARNING ENVIRONMENTS

Guy Groen
McGill University
Department of Educational Psychology

Most discussions of the relevance of AI to CAI have focussed on some aspect of systems development. This paper will deal with a different topic: protocol analysis and its potential relevance to the development of more useful and realistic evaluation techniques in informal learning environments involving unstructured gaming or open-ended problem solving. Straightforward simulation programs are the most obvious examples. Much of intelligent CAI also falls into this category, as do much freer environments based on programming languages such as BASIC, LOGO and SMALLTALK.

Such environments have extremely subtle objectives that resist standard evaluation techniques. For example, the claim is frequently made that powerful skills and heuristics are being taught that transcend the immediate task environment. The proper examination of these objectives is extremely important. As Burton and Brown (1978) point out, an informal environment is one that the student is able to explore at will, free to create his own ideas of underlying structure and to invent his own strategies for utilizing his understanding of this structure. If properly designed, it can lead to the formation of strategies and knowledge structures that have general usefulness in other domains. On the other hand, a major stumbling block to the effective use of such an environment is the distinct possibility that the student will form a grossly incorrect conception of the underlying structure, be unaware of the limits of his strategies and be unable to discover the causes of his overt errors or "bugs". I am calling an approach to evaluation that takes such matters into account "intelligent" in order to emphasize that notions of artificial intelligence seem destined to play as large a role in its development as it is currently playing in the development of the environments themselves.

Some Limitations of Standard Evaluation Techniques

Most standard evaluation techniques are atheoretial and are based on pretest-postest comparisons that virtually ignore the student's interaction with the computer. A study conducted by myself in collaboration with John Kreeft and Evie Tsouna (Groen, Kreeft and Tsouna, 1979) serves to illustrate some of their limitations.

This research began with a program that simulated a patient suffering from aspirin poisoning. The student's task was to "cure" the "patient" by administering appropriate drug therapy over a period of time. The program was based on an explicit dynamic model of the patient's physiological functioning, so there was some reason to believe that it could be embedded in a tutorial context using some of the techniques of Intelligent CAI. The issue was whether this was worth the effort from a practical point of view. Any attempt at Intelligent CAI is a major research undertaking. In contrast, tutorial frame-oriented CAI is conceptually straightforward and quick to implement. Hence, it seemed desirable to begin by examining what would actually be learned in this more conventional form of instruction. The most reasonable hypothesis is that traditional CAI teaches factual knowledge whereas simulation (even without a tutorial component) teaches high level heuristics. Specifically, traditional CAI teaches the basic pharmacology whereas simulation teaches the clinical skills required for successful therapy. The experiment was designed to test this hypothesis.

A conventional text oriented program was implemented that was designed to teach the same basic material as the simulation, (as much as could be verbalized). Both it and the simulation program were designed by the same individual (a clinical pharmacologist with extensive experience in computer programming). Two test forms were designed containing two classes of items: "basic" items designed to elicit knowlege of basic pharmacology, and "clinical" items designed to elicit knowledge of clinical treatment. A conventional counterbalanced design was used. 54 second year and 50 fourth year McGill medical students were used as subjects. Each received a pretest followed by one of the two program followed by a post-test administered immediately after completing the program.

None of the pretest differences were statistically significant. Neither were the various differences between pretest and post-test for the simulation program. On the other hand, the conventional tutorial program yielded significant pretest - post-test differences for both 'clinical' and 'basic' items. The post-test differences between the tutorial and simulation condictions were also superior for both classes of items.

It seems safe to conclude that the conventional tutorial program is clearly superior to the simulation program, as far as the kind of learning measured by these tests is concerned. Indeed, there is no evidence that anything at all is being learned from the simulation program. This appears surprising, in the light of the current popularity of pure simulation programs, at least in the field of medicine. However, a number of alternative explanations are possible. The most appealing is simply that the tests are not measuring the actual skills, but rather the ability to process verbal information about them. This leaves us in a peculiar situation. The most thorough way to measure such skills (barring actual observation in a clinical setting) is to simulate the clinical situation. In other words, we must use the simulation to evaluate itself!

All this serves to illustrate the basic shortcoming of the standard pretest-postest approach to evaluation where informal learning environments are concerned. We are not looking at the basic strategies and knowledge structures themselves. Instead, we are looking at transfer of training from the environment to some other task. When what is learned in the actual environment is ill-defined, tests for transfer of training can only be developed in an ad-hoc fashion, and the obtaining of interpretable effects is likely to be difficult indeed.

The precise definition of such strategies and knowledge structures is equivalent to the development of a theory of how and what the student learns.

Evaluation is equivalent to testing the theory. It seems reasonable to assert that such a theory can only be adequately tested through the use of data emanating from the student's interaction with the computer.

Protocol Analysis

There are two ways of developing theories about informal learning environments. One can proceed in a top-down fashion, using the theory as a means of structuring and classifying the data. This is quite feasible in highly evolved Intelligent CAI systems (e.g., Burton & Brown, 1978) in which a theory is already present as an essential component of the functioning system. It has also been used in more open-ended situations (Miller & Goldstein, 1977). It has the advantage of yielding a coherent picture of what the student is doing. However, it involves commiting oneself in advance to a rigid data base. As an environment becomes more unconstrained, the danger increases that a pre-defined data base will miss important aspects of the students' behavior.

In this paper, I will consider a more bottom-up approach that is essentially a form of protocol analysis. This evolved in the course of some research, conducted in collaboration with Glenn Cartwright and Sharnee Chait, which was concerned with a graphics-oriented environment based on the programming language LOGO (Chait, 1978). The system consists of a DEC LSI/11 computer connected to a special purpose system which generates graphics on a cathode ray screen. LOGO is a simplified version of LISP designed to be usable by children (Papert, 1972). It contains commands that enable an arrowhead on the screen to move in a straight line for a specified distance, and to turn in a specified direction. The arrowhead can be commanded to leave a trace on the screen of where it has been. Quite complex pictures of such things as geometrical objects, houses, faces and vehicles can be generated by programs utilizing these commands.

This is a free learning environment that has aroused considerable controversy. Its developers have had ambitious goals concerning the improvement of childrens' thinking. For the same kinds of reasons as

I have already discussed, standard evaluation procedures have been unsuccesful in establishing whether or not children actually attain these goals. The purpose of our research was not to conduct anything remotely resembling an evaluation study. It was simply to explore whether we could isolate anything of interest about the strategies and knowledge structures of children free to explore this environment by collecting and analyzing an extremely detailed set of protocols.

The subjects were 5 students in Grade 7 with low mathematics grades, and no previous experience with computers. Each student participated individually in 2 one-hour sessions each week over a period of 3 months. Projects were self-selected. At the beginning of each project, the student was asked to make a drawing on a piece of paper, of whatever he intended to construct. He then proceeded to write an appropriate set of procedures and subprocedures to attain this goal. Students were free to modify or abandon a project whenever they wished. They were encouraged to go on to a new project whenever they felt they had done all they wanted with the current one. All students successfully completed a number of such projects. The basic raw data consisted of the so-called dribble file, a complete record of the student's input and the computer's output. This was supplemented by a detailed record, prepared by the experimenter immediately after each session, of whatever the student said or did as he/she was interacting with the computer.

Probably the most highly evolved technique currently available for the analysis of human problem solving protocols is the method developed by Newell and Simon (1972). Typically, the subject is given a problem to solve and is asked to "think aloud" as he is solving it. Roughly speaking, the analysis of the ensuing protocol proceeds in three phases. First, the protocol is divided into a sequence of episodes (each episode corresponding to something new that the subject does). An attempt is then made to develop a uniform scheme for characterising the subject's processes in each episode. Finally, a model is developed that characterises the strategies and knowledge structures used by the subject in solving the problem. Unfortunately, this technique suffers from two disadvantages when applied to a situation of the type considered here. The first is that "thinking-aloud" protocols are difficult to obtain without interfering with the teaching process. The second is that the analysis itself is extremely time consuming. Newell and Simon's technique was developed in the context of experiments involving a single subject solving a problem assigned by the experimenter. The result was what might be called a micro-analysis: an attempt to characterise everything of interest in an extremely detailed protocol. What was necessary for our purposes was a method that could be applied as uniformly as possible to the less detailed protocols of several subjects solving problems of their own choosing.

The approach that we developed might best be called a macro-analysis. Newell and Simon's notion of dividing the protocol into episodes was followed. However, the episodes were of much longer duration (between 5 and 15 minutes). What each episode determined in this way seemed to capture was a change in a high-level goal (in contrast, the approach of Newell and Simon would have attempted to capture changes in goals at all levels). A uniform classification scheme was then developed that characterised the more overt aspects of what each child was doing in each episode. The result was a set of summary protocols that allowed comparisons to be made both between projects and between subjects.

These comparisons yielded two interesting results. The first is that subjects appear to cycle through a series of stages in developing a project; (a) a planning stage corresponding to the construction of a high level goal; (b) a stage which, for want of a better term, we call generate-and-test, which essentially consists of the generation and testing of highly specific local hypotheses about how to change the goal; (c) writing the actual procedure; (d) editing the procedure. It should be noted that there is no debugging stage. Most non-trivial debugging appears to occur in stage b. The second is some evidence for two kinds of "programming style". The links between episodes in a given project can be represented in the form of procedural graph. Such graphs tend to be of two types. In the first, the links are mainly those that denote connections between sub-procedures and super-procedures. In the second, the vast majority denote revisions of procedures that possess little in the way of hierarchical structure.

Clearly, this work stops far short of the type of analysis performed by Newell and Simon. However, it does show that it is possible to analyze large numbers of protocols in such a way that interesting uniformities and differences in subjects' behavior begin to be revealed. The further analysis of these uniformities, together with a deeper and more precise characterization of the processes and knowledge structures that are involved, might ultimately lead to the disovery of what (if anything) it is of importance that a child learns from LOGO. Moreover, such results might generalise to other free environments based on programming languages.

References

Burton, R.R. & Brown, J.S. An Investigation of Computer Coaching for Informal Learning Activities. To appear in: Brian Gaines (Ed.) International Journal of Man-Machine Studies, 1978

Chait, S. An Analysis of Childrens' Problem Solving in a Graphics Oriented Computer Programming Environment. M.A. thesis, McGill University, 1978.

Groen, G.J., Kreeft, J. and Tsouna, E. A Comparison Simulation with Traditional CAI in a Medical School Setting. Paper delivered at AERA 1979.

Miller, M. & Goldstein, I. Structural Planning and Debugging. Proceedings of International Joint Conference on Artificial Intelligence, 1977.

Newell, A. & Simon, H.A. Human Problem Solving. Prentice Hall, 1972.

Papert, S. Teaching Children Thinking. Programmed Learning and Educational Technology. September, 1972. 245-255.

Socratic Teaching of Causal Knowledge and Reasoning

Madeleine Bates
Geoffrey Brown
Allan Collins

Bolt Beranek and Newman Inc.
Cambridge, Massachusetts

Introduction and Background

Cognitive science provides a set of theoretical formalisms and analysis techniques that can be used to study teaching and learning [Collins, 1978]. All too often good human teachers cannot verbalize their teaching strategies (and therefore cannot teach other teachers effectively) because the application of those strategies requires only an intuitive feel as to how to use them. Now that computers have provided us with the capability of expressing and experimenting with process models, we can begin to build, extend, refine, and communicate theories of instruction and learning. The computer system described in this paper, the WHY system, is an attempt to explore the factors involved in skillful Socratic tutoring of causally related physical processes.

The Socratic method [Collins, 1977; Plato, 1924] involves systematically asking an appropriate sequence of questions in order to lead the student to discover important relationships between pieces of knowledge which were previously known but whose interaction had not been consciously explored. During this process, the student is exposed to a process of reasoning which the student can attempt to follow again without the aid of the tutor. Clearly, this type of teaching embodies a set of complex skills including reasoning, knowledge of the subject, knowledge of the student, a plan for teaching, language production, and language comprehension. Each of these facets must be highly developed (incompetence in any of them would result in a frustrating, unsuccessful lesson) and each of them is a major research area in cognitive science and/or artificial intelligence.

At the beginning of this project, tutorial sessions about global weather patterns using human tutors and students were recorded, transcribed, and analyzed [Collins, 1977; Stevens and Collins, 1977]. The analysis revealed that tutors discuss topics in a rational order, typically following the discussion of one process (such as evaporation or current movement) with a discussion of a temporally or causally adjacent process or with a discussion of a subprocess. When the student makes an error, the tutor may either correct it immediately or suspend the current line of questioning to focus on other questions which lead the student to isolate and recognize his/her misconception. When a conceptual bug is diagnosed, the tutor attempts to correct it. This may require a simple statement of fact, an additional question, or an extended dialogue during which the tutor continues to probe and modify the student's knowledge. Tutors may also probe the student's knowledge of facts and reasoning skills by asking for predictions in selected cases (e.g. "What happens to the moist air after it moves to Ireland?"), and by trying to entrap the student into making incorrect predictions. Tutors' comments about the dialogues revealed that they used information about common student errors and their estimation of what the student knew (i.e., a student model) in order to formulate their questions.

The WHY System

Three main parts comprise the WHY system: a tutor, a dialogue controller, and a natural language analyzer and generator. The most prominent data sources are a process representation of the causal chain of events to be taught, a student model, and a set of goals with strategies for achieving them. Figure 1 shows a schematic of the organization of the system. Functional components are represented by boxes and data sources by ovals; data structures which are passed from component to component are labeled. The various parts of the system will be described below, beginning with the process model, and then their synthesis will be examined. This section ends with an annotated example of a dialogue with the WHY system; the reader may wish to refer to it during the following discussion.

The physical process of rainfall was chosen as the subject to be taught by the WHY system because most people know quite a lot about parts of the process

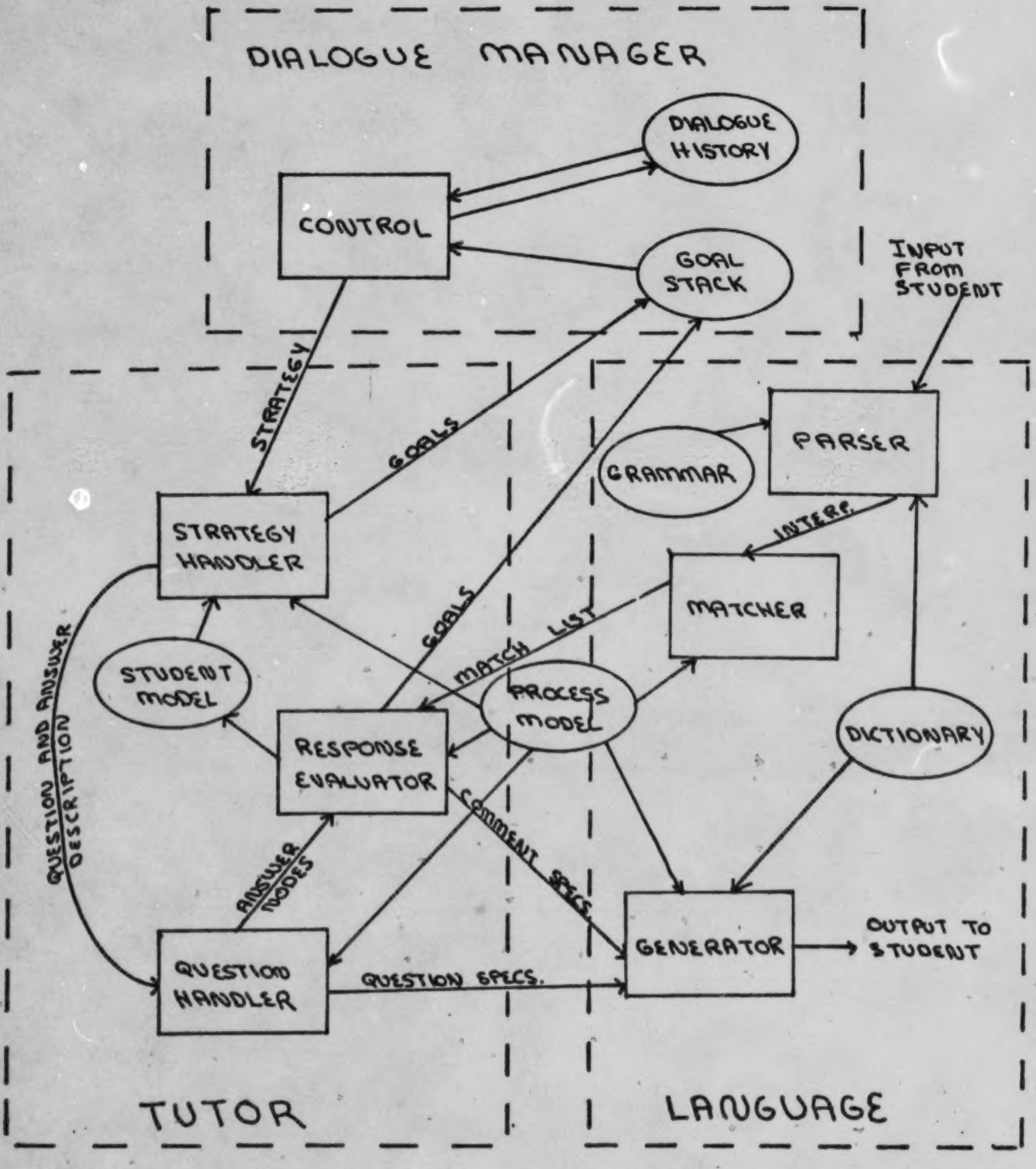

DIALOGUE MANAGER
DIALOGUE HISTORY
CONTROL
GOAL STACK
INPUT FROM STUDENT
STRATEGY
GOALS
PARSER
GRAMMAR
STRATEGY HANDLER
INTERP.
MATCHER
GOALS
MATCH LIST
QUESTION AND ANSWER DESCRIPTION
STUDENT MODEL
PROCESS MODEL
RESPONSE EVALUATOR
DICTIONARY
COMMENT SPECS.
ANSWER NODES
QUESTION HANDLER
QUESTION SPECS.
GENERATOR
OUTPUT TO STUDENT
TUTOR
LANGUAGE

Figure 1. A Structural Overview of the WHY System

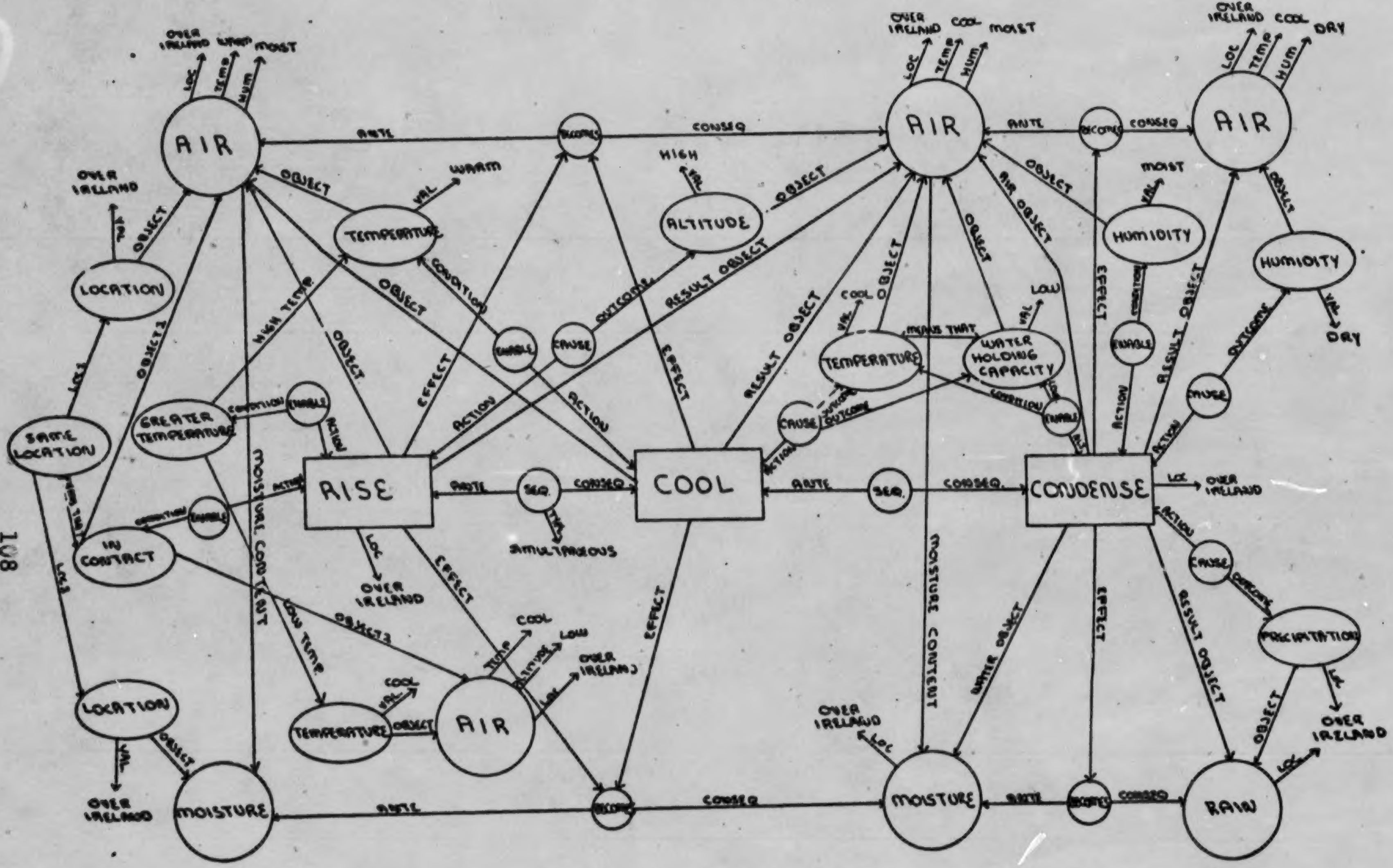

OVER IRELAND
WARM
MOIST
LOC
TEMP
HUM
AIR
ANTE
BECOMES
CONSEQ
AIR
LOC
TEMP
HUM
OVER IRELAND
COOL
MOIST
ANTE
BECOMES
CONSEQ
AIR
LOC
TEMP
HUM
OVER IRELAND
COOL
DRY
OBJECT
VAL
WARM
TEMPERATURE
HIGH
VAL
ALTITUDE
OBJECT
OBJECT
AIR OBJECT
HUMIDITY
MOIST
VAL
HUMIDITY
OBJECT
VAL
DRY
OVER IRELAND
VAL
OBJECT
LOCATION
LOC 1
OBJECT 1
HIGH TEMP.
OBJECT
OBJECT
CONDITION
ENABLE
CAUSE
OUTCOME
RESULT OBJECT
EFFECT
EFFECT
ACTION
ACTION
RESULT OBJECT
OBJECT
COOL
VAL
TEMPERATURE
MEANS THAT
WATER HOLDING CAPACITY
LOW
VAL
CONDITION
ENABLE
ACTION
CAUSE
OUTCOME
OUTCOME
EFFECT
CONDITION
ENABLE
ACTION
RESULT OBJECT
CAUSE
ACTION
OUTCOME
SAME LOCATION
GREATER TEMPERATURE
CONDITION
ENABLE
ACTION
RISE
ANTE
SEQ.
CONSEQ
TYPE
SIMULTANEOUS
COOL
ANTE
SEQ.
CONSEQ.
CONDENSE
LOC
OVER IRELAND
ACTION
CAUSE
OUTCOME
PRECIPITATION
LOC
OVER IRELAND
IN CONTACT
CONDITION
ENABLE
ACTION
LOC 2
MOISTURE CONTENT
LOC
OVER IRELAND
EFFECT
OBJECT 2
LOW TEMP.
EFFECT
MOISTURE CONTENT
WATER OBJECT
EFFECT
RESULT OBJECT
OBJECT
LOCATION
VAL
OVER IRELAND
OBJECT
TEMPERATURE
COOL
VAL
OBJECT
AIR
TEMP
COOL
ALTITUDE
LOW
LOC
OVER IRELAND
MOISTURE
ANTE
BECOMES
CONSEQ
MOISTURE
OVER IRELAND
LOC
ANTE
BECOMES
CONSEQ
RAIN
LOC
OVER IRELAND

Figure 2. A Portion of the Process Model Network

The information in the student model is used by the tutor to determine what questions to ask and how to phrase them. For example, the tutor would not ask a question if the student has already demonstrated knowledge of the answer unless the tutor intends to review material. Similarly, if a student has exhibited knowledge of some, but not all, of the preconditions of a concept, the tutor might ask "What else causes that?".

Key elements in the automation of the tutoring process are the selection and representation of tutorial strategies and goals. A goal is an objective such as choosing a particular case of a general process, expanding the student's knowledge to prior and/or subsequent factors of an event, correcting the student, or questioning the student about the outcome of a process or the next process in the causal chain. A strategy is a procedure designed to achieve a certain goal. For example, the strategy for exploring prior factors of an event involves checking the process model to determine that the prior factors are known by the system, eliminating those already known by the student, generating a question about one of the remaining factors, and creating a new goal to investigate the others. As a strategy is executed, additional goals are often generated and are placed on the system's goal stack. When a goal reaches the top of the stack, the strategies for achieving it are retrieved and one is chosen and run. In addition to placing new goals on the stack, a strategy may have other effects such as modifying the student model and calling the sentence generator to produce output for the student. Because the student's response to a question is processed in light of the strategy which has as its goal soliciting the correct answer to the question, the tutor has control over what to do if the student answers incorrectly or partially. We will return to this problem, but first it is necessary to investigate the language processing components of the system.

The input side of the natural language "front end" for the system is an augmented transition network (ATN) parser which uses a very sophisticated and general ATN grammar together with a dictionary of more than 1000 words containing, in addition to the usual syntactic features, syntactic and semantic case information [Webber and Bobrow, 1978]. With this case information the parser is able to construct a semantic interpretation of the input string as it is parsed, thus never producing a syntactically correct but semantically meaningless interpretation. The availability of semantic information which can be used to screen out ambiguities also greatly increases parsing speed. The grammar can handle complex relative clauses, complements, comparatives, passives, questions, extrapositions, ellipsis, gapping, and conjunction, although not all of them can presently be handled in their most general form. Several examples of the case-structured output produced by the parser are given in Figure 3.

But producing such an interpretation is less than half the job of processing a student's response. It is also necessary to determine what portion of the process model is embodied by the student's answer and how closely that resembles a correct answer to the question. The first of these tasks is performed by a part of the system called the matcher, and the second by the response evaluator.

The matcher is given the student's parsed answer and a set of nodes in the process model network against which to match the interpretation. It is important for efficiency's sake that this set not be the entire network, but it is equally important that not just the "right" answer be given, as that would prejudice the matcher toward finding the right answer in the student's response even if there were only a small correlation between them. A set of nodes embodying, say, the specific case of rainfall being discussed appears to be the right size to allow erroneous answers to be matched without sacrificing efficiency.

The output of the matcher is a set of nodes which the answer appeared to match, ordered by the degree of completeness of the match, together with indications of which related nodes were actually mentioned and a list of errors. For example, the interpretation of "the cool moist air" would match the node embodying "the warm moist air over Ireland" with annotations that the location of the air was omitted, that the humidity was specified, and that the temperature was given the opposite value; it might match other nodes as well, such as the node for "the cool air over Ireland". The response evaluator, which is initialized by the

Sentence 1: A lot of moisture is absorbed by the warm
air over the Gulf Stream.

```
[DECLARATIVE
  SUBJECT = [NP
              LOCATION = [PP
                           LOC = [NP
                                   NAME = GULFSTREAM
                                   FRAMETYPE = <CURRENT>]
                                REL = OVER
                           FRAMETYPE = <LOCATION>
                           PREP = OVER]
              TEMPERATURE = HOT
              FRAMETYPE = <AIR>]
  OBJECT = [NP
             QUANTITY = HIGH
             FORM = MOISTURE
             FRAMETYPE = <WATER>]
  FRAMETYPE = <ABSORPTION>]
```

Sentence 2: The moist air is blown from the equatorial
Atalntic to Ireland.

```
[DECLARATIVE
  AGENT = [NP
            FRAMETYPE = <WIND>
            IMPLICIT = T]
  DESTINATION = [NP
                  FEATURE = COUNTRY
                  NAME = IRELAND
                  FRAMETYPE = <LOCATION>]
  SOURCE = [NP
             LOCATION =[LOCATION
                         NAME = EQUATOR]
             NAME = ATLANTICOCEAN
             FRAMETYPE = <OCEAN>]
  OBJECT = [NP
             HUMIDITY = HIGH
             FRAMETYPE = <AIR>]
  FRAMETYPE = <MOVE>]
```

Figure 3. Output from the Parser

currently active strategy with the node or nodes of the network comprising the right answer and which is given the results of the matcher, must determine how well the answer fits what was expected. This important step cannot be overlooked, nor can it be performed by the matcher; a response of "California" matches one node of the network perfectly, yet is the wrong answer to "Where does the Gulf Stream flow?". The incorrect portions of an answer and the omitted portions of a partial answer provide the response evaluator with a wealth of information to be used in formulating subsequent goals, although the current implementation is rudimentary. The response evaluator updates the student model, calls the sentence generator to make comments on the student's answer, and decides which new goals to add to the system's goal stack.

The language generator is driven from the process network with a set of specifications provided by the strategy calling the generator. Each node has associated with it a set of generation templates which show how to express that node in a variety of syntactic forms. Part of the template may call for the recursive generation of another node which is related to the first one.

An example of the generation templates for the concept of evaporation is shown in Figure 4. The concept may be syntactically realized as a sentence, a noun phrase, a noun, a verb, or a short sentence. "Opt" is an indication of optionality. "Pick" is a selector which randomly chooses among alternatives. For example, a sentence could be generated as either "The water evaporates into the air" or "The air absorbs the water". Recursive calls to generate constituents from other nodes are indicated by the forms (Using <link name> <syntactic form>) and (Using nodE <node name> <syntactic form>): the former follows the given link from the current node to another one which is generated according to the specifications in the syntactic form, the latter uses the named node in the same way. Thus depending on which node was at the end of the Source link, the subject of the sentence might be generated as "the water in the Gulf Stream" or "the warm water at the equator".

A specification to the generator may be a requirement for a specific syntactic form (e.g. a

```
<Evaporate>  (S (pick ((Using Source (NP *))
                       (V *)
                       (Using Object (PP (Prep into)
                                         (NP *))))
                      ((Using Object (NP (with no Location)
                                         *))
                       (UsingNode Absorb (V *))
                       (Using Source (NP *)))))
             (NP
               (pick
                 ((Art the)
                  (N evaporation)
                  (opt (Using Source
                              (PP (Prep of)
                                  (NP (with (word from)
                                            -- Prep)
                                      *))))
                  (opt (Using Object
                              (PP (Prep into)
                                  (NP (with no Location)
                                      *)))))
                 ((S (Using Source (NP (with (word from)
                                             -- Prep)
                                       *))
                     (V (with (VForm NIL NIL ProgressiveForm))
                        *)
                     (opt (Using Object
                                 (PP (Prep into)
                                     (NP (with no Location)
                                         *))))))))
             (N evaporation)
             (V MakeVerbForm)
             (Verb evaporate)
             (ShortS (S (pick ((NP (N *))
                                 (UsingNode Occur (V *)))
                                ((Using WaterObject
                                        (NP (with no -- Adjs)
                                            (with no Location)
                                            *))
                                 (V *)))))
```

Figure 4. Templates for Language Generation

question of a given type, a noun phrase, a declarative sentence, an adjective), an indication that certain information is or is not to be generated (e.g. express the node for "the cool dry air low over Ireland" without mentioning the temperature), or a request for a given level of complexity in the output. The complexity level is the beginning of an attempt to tune the system's output to the degree of sophistication and knowledge shown by the student. For example, one might ask a very bright student "What happens when the water evaporates?" while another student may need to have the focus of the question made more obvious ("What happens to the air when the water evaporates"), and yet another may need a strong hint ("What happens to the humidity of the air when the water evaporates?").

The generator has a rather naive capacity for generating pronouns and for ellipsing material which has been mentioned previously. Careful specification of the restrictions to be observed when generating combined with random selection among some alternatives results in very readable, though somewhat formal, output.

The following annotated transcript shows the WHY system in operation; the comments in brackets describe what goes on inside the system to produce the questions and process the responses.

> {The system begins with a prestablished goal of selecting a case. The strategy for that goal selects the case of heavy rainfall in Ireland and generates a goal of introducing the subject to the student. The introduction strategy generates a brief introduction and asks the first question.}

WHY system: We will be discussing why rainfall occurs in Ireland. Ireland has fairly heavy rainfall. In order to have heavy rainfall there must be a lot of moisture in the air. For Ireland the moist air comes from over the Gulf Stream current. How does moisture get into the air over the Gulf Stream?

Student: It evaporates

{The response parses and matches the node for evaporation of water from the ocean into the air. This node is marked in the student model as known. The next goal to be generated is to expand the student's knowledge to prior causes, and since the student has not indicated knowledge of any of the prior conditions, all are looked for by generating a question about the conditions enabling the process of evaporation}

WHY system: Correct, why does the air mass absorb the water in the Gulf Stream?

Student: The air is warm

{This response matches one of the enabling conditions of evaporation, but others have yet to be mentioned so a goal is generated to ask about the rest of the enabling conditions.}

WHY system: Correct, what else makes it possible for the water to evaporate?

Student: The water must be warm

{This response matches the last of the enabling conditions so the goal of determining what causes evaporation has been satisfied. To continue the goal of exploring the student's knowledge of prior processes, a subgoal is generated to ask about the antecedents of the air mass.}

WHY system: Yes that's right, where does the warm air over the gulf stream come from?

Student: I don't know

{This vacuous response does not match any nodes in the net, so the tutor decides to inform the student of the right answer to the question.}

Comment: The prevailing wind blows the air mass from over the Equatorial Atlantic to over the Gulf Stream.

Problems and Limitations

It should be obvious that the system is, and must be, composed of a large number of complex interacting pieces, all of which must work extremely well for the system to perform in a manner which appears intelligent.

One source of difficulty is student responses which are either extremely ungrammatical, semantically meaningless, or simply beyond the knowledge capacity of the system. (One student typed something like: "I think it rains a lot in Ireland because Ireland makes me think of shamrocks and they need a lot of water.") Of course such responses can always be ignored, but it would be far better if the system could partially comprehend such inputs. Other language processing stumbling blocks include the use of pronouns, anaphora, conjunction, and stylistic variation, to name just a few.

It is important to emphasize, however, that the language problems are by no means the most serious impediments to an intelligent system. Another problem is that of providing integrated strategies and a structure to drive the interactions to produce a smoothly flowing dialogue. The initial control structure which was used, a production rule system, created goals without much continuity between interactions. The current modified top down approach is a considerable improvement because it combines a goal driven approach with the ability to modify the course of the dialogue by reacting to the student's input, but further refinement would be desirable. There is much more to be learned about strategies of ordering the presentation of material, focusing the dialogue on particular issues, planning focus shifts, and making those shifts smoothly. Particular strategies such as generating counter-examples also need considerable investigation before they can be implemented satisfactorily.

Other problems occur when a student makes an incorrect answer which matches part of the network other than the correct answer: Is the match just accidental or does it really represent what the student meant? If a pronoun reference is ambiguous, should the assumption be made that the student was correct or should further questions be asked for clarification? When the student has given a sequence of incorrect answers, how can the system determine whether that is the result of total confusion or whether the student is deliberately making bad responses to test the system? When the student has almost the right answer but brings in irrelevant material (such as clouds), how can the tutor accept the answer as correct yet inform the student of the irrelevant issue?

Another severe limitation is the fact that the current student model is unable to represent incorrect information. If the student really believes that water is the Gulf Stream is cold or that clouds absorb water, the ability to represent those beliefs and to infer their consequences would aid the tutor in developing counter-examples and entrapments. But if such erroneous information were represented in the student model, perhaps using partitioned networks to implement belief spaces, a host of other problems would be raised: How should the matcher and response evaluator handle such nodes? When should they be created? When should the generator use them? How would the problem of truth maintenance be handled when inferring from "bad data"?

It is very instructive to compare the WHY system to the SOPHIE system [Brown and Burton, 1975] which can apparently handle a higher percentage of student responses correctly. SOPHIE gains its advantage from two facts. First, SOPHIE's universe of discourse is that of trouble-shooting a fault in a simple electronic circuit; this domain is quite closed in terms of both the sorts of questions and statements can be made about it and the limitations on inferences which must be made by the system. In addition, the kind of language employed by users of SOPHIE is so limited that a simple semantic grammar could be used. The rainfall domain of WHY, on the other hand, is far more open and invites both a wider range of syntactic constructions and a broader set of concepts.

Conclusion

The WHY system, while far from ready for use by actual students, is an excellent vehicle for research in natural language processing and intelligent computer aided instruction. We hope that this research has not only shown what is achievable within the limits of current technology, but also indicated the directions in which the state of the art must be extended.

Bibliography

Brown, J.S., Burton, R.R. and Bell, A.G. SOPHIE: A Step Towards a Reactive Learning Environment, International Journal of Man Machine Studies, 7(1975) p. 675 - 696

Collins, Allan Explicating the Tacit Knowledge in Teaching and Learning, paper presented at the American Education Research Association (also BBN Report 3889), March 1978

Plato. Laches, protagoras, Memo, and Euthydemus (WRM Lamb, trans.) Cambridge, Mass., Harvard University Press, 1924. Stevens, A. and Collins, A. The Goal Structure of a Socratic Tutor. In Proceedings of the ACM Annual Conference, 1977 (also BBN Report no. 3518)

Stevens, Albert L., Collins, Allan, and Goldin, Sally, Diagnosing Students' Misconceptions in Causal Models, BBN Report 3786, June 1978

Webber, B.L. and Bobrow, R. Research in Natural Language Understanding; Quarterly Technical Progress Report no. 3, 1 March 1978 to 31 May 1978, (BBN Report no. 3878, July 1978)

Toward the Development of Intelligent CAI*

William S. Bregar
Computer Science Dept.
Oregon State University

Arthur M. Farley
Computer Science Dept.
University of Oregon

ABSTRACT

A model of an intelligent CAI system is described. The model is based on the concept of interactive problem solving. A problem solver (the operative subsystem) based on means-ends analysis interacts with an understander (the descriptive subsystem) which can interpret the steps the operative subsystem takes in solving a problem.

The operative component consists of a production system, the descriptive component is incorporated in a problem solving grammar. The system currently focuses on the domain of high school algebra problems. By monitoring and understanding the problem-solving activities the descriptive subsystem is capable of providing relevant suggestions to the operative subsystem to enhance its problem solving ability. Viewing the operative subsystem as a student, one can envision the model as the basis of a CAI system for teaching problem solving in a variety of domains.

Introduction

In the past several years there has been a steady increase in the development of Intelligent CAI (ICAI) systems [1, 2, 4, for example].

Each of these systems exhibit certain characteristics in the context of the traditional view of CAI: (1) the systems are developed around a fairly narrow range of subject matter - thus they do not represent courseware in the sense of a term's worth of material; (2) they often allow a more sophisticated level of natural language interaction between the computer and the student; (3) they are capable of building detailed models of student behavior in problem solving tasks; (4) the

---

This work was partially supported by grant #78-0093, National Institute of Education.

systems are generally experts in solving the class of problems they present to students. The latter two features combine to drive the tutorial functions of these systems which is manifest through the natural language *link mentioned in point 2.*

Information Processing Models of Intelligence

Artificial Intelligence spans at least three disciplines - Psychology, Linguistics and Computer Sciences. Its breadth prevents more than a superficial discussion of any specific area, therefore, we will focus one particular aspect of AI research - models of intelligence. We introduce the concept of an *information processing* theory of intelligence. Such a theory is usually stated as a computer program in a formal programming language and postulates in detail a precise set of mechanisms to account for the observed behavior [6]. Such mechanisms would include both procedural and representational elements which combine to produce or predict behavior in a variety of contexts. A model of memory, for example, might employ semantic networks as a representation along with a set of mechanisms for integration of new information, retrieval of existing information, and some procedures for performing simple inferences.

One indication of a system's intelligence within a given domain is its ability to perform problem solving tasks in that domain. Success on such tasks may result from an operation as simple as a direct retrieval of the solution from the system's knowledge base. On the other hand, successful problem solving may depend upon the development and execution of a sophisticated plan [8]. Such plans are usually developed on the basis of some underlying strategy.

Polya [7] and Wickelgren [9] have both written of the types of strategies involved in mathematical reasoning. Often their strategies are implemented in the form of heuristics which guide the problem solving process. One particularly effective heuristic method employed in a variety of AI problem solving systems is *means-ends analysis* [3]. In using this heuristic a problem solver applies one of a set of available operations in an attempt to reduce or eliminate its perception of the *difference* between the current situation and the desired (or goal) problem state. Problem solving operations are chosen on the basis of these differences, hence the term means-ends analysis. A wide variety of problems ranging from logical puzzles such as missionary and cannibal problems to water jug problems have been solved using means-ends analysis.

Implications of AI Problem Solvers for CAI

Programs which are capable of solving problems utilizing a strategy such as means-ends, can be employed in a tutorial CAI environment. By a tutorial environment we refer to a system which presents a problem to a student and monitors his/her progress towards a solution. The system may comment or query the student on the methods he/she is using. If the student encounters difficulties, the system can suggest appropriate alternatives based on its understanding of the approach the student has taken so far. Thus, the system must be capable of determining from a student's set of inputs the likely problem solving methods or strategies being employed. Suggestions from the system can therefore be based upon the student's own approach to the problem if it is perceived as being a valid one. Strategy shifts can also be recommended in the event that the system concludes that the student is following a dead-end path.

Towards this end, we have developed a model for the interactive solving of high school algebra word problems. We focus on the processes by which a problem solver moves from the statement of the problem to appropriate equations which can then be solved using standard techniques. This process is modeled using a production system [3] which incorporates means-ends analysis to solve problems and is referred to as the operative subsystem. The performance of the operative subsystem is monitored by a descriptive subsystem which creates from the sequence of operations it sees a hierarchical structure (tree) wherein the higher levels could be said to account for the appearance of elements at the lower levels. In essence, the descriptive subsystem acts as a parser of the sequence of operations from the production system to ascribe a structure to a problem solution. The descriptive subsystem, in fact, is a problem solving grammar akin to those of Miller and Goldstein [5]. The grammar consists of a set of high level or general problem solving rules and a replaceable group of lower level domain specific rules. For high school algebra word problems, the low level rules account for the algebraic manipulation of equations or sets of equations leading to the isolation and subsequent assignment of a value to a variable. The higher level rules recognize action sequences concerned with the establishment of a problem context, gathering of information with respect to a goal, and generation of appropriate subproblems based on a means-ends strategy.

To give the reader a greater appreciation for the proposed theory, we present the higher level rules of the descriptive subsystem and apply them to an example problem.

The operations we propose for solving word problems range from general problem solving tactics to specific algebraic manipulations. This provides us with a hierarchic approach to problem solving, where within each level there exists a prescribed set of possible sequences of activities. Furthermore, the lower level can be modified as necessary to reflect differences in the problem domains.

Specifically, the general (top level) activity of problem solving consists of IDENTIFICATION of the problem, SOLVING it and EMPLOYING the results. We represent this in the form of a rule:

PROBLEM SOLVING → IDENTIFY SOLVE EMPLOY

The elements to the right of the arrow break down the element on the left into more elementary operations which could be said to constitute the left hand element. Each of these may, in turn, be broken down arbitrarily far. At the lowest level primitive actions are specified which combine to perform the operations of which they are constituents.

For example, IDENTIFY in the rule above is the left side of the rule

IDENTIFY → DETERMINE-CONTEXT DETERMINE-GOAL

Completing the sequence

DETERMINE-CONTEXT → read-problem
DETERMINE-GOAL → note-goal-variable

The small letters designate the lowest level to which a rule can go.

The rules in Figure 1 comprise the basis for problem solving behavior in a wide variety of contexts. These rules provide for the identification of the problem and a goal for which to solve, the acquisition and subsequent processing of relevant information, and the generation of appropriate subproblems.

1. PROBLEM → IDENTIFY SOLVE EMPLOY-RESULT
2. IDENTIFY → DETERMINE-CONTEXT DETERMINE-GOAL
3. SOLVE → ACQUIRE-INFO PROCESS-INFO* NOTE-RESULT
4. ACQUIRE-INFO → PERCEIVE|RECALL|REACTIVATE
5. PROCESS-INFO → MANIPULATE*|SELECT-PROBLEM*
6. SELECT-PROBLEM → OLD PROBLEM|SUBPROBLEM
7. SUBPROBLEM → DETERMINE-SUBGOAL SOLVE EMPLOY-RESULT

Note: Each rule specifies a sequence (1 or more) of problems solving operations. An asterisk means that the specified operation may be performed an arbitrary number of times. A vertical bar is read as "or", and separate the alternatives for a particular rule.

Figure 1

Now, consider the following problem:

> Tom has twice as many peaches as Mary. Mary has 6 more peaches than Fred. Fred has two peaches. How many peaches does Tom have?

Note that in the following we show only one of several possible ways each problem could be solved within the framework of the theory. In fact, one value of the theory is its ability to account for a variety of solution paths. The descriptions are presented in terms of the steps a student might take when utilizing the different detection mechanisms and operations of the theory.

Faced with the example problem, the student begins by IDENTIFYing the problem, establishing a context and determining the relevant goal or goals. The context consists of the information and question elements in the problem statement; thus from the question "How many peaches does Tom have?", Tom* is designated as the goal variable.

The next step is to SOLVE the problem by finding a value for the goal variable. The ACQUIRE-INFO component of rule 3 is utilized to gather available information about TOM. The only source of information pertaining to TOM is the problem context, a search of which yields the information

$$\text{TOM} = 2 \times \text{MARY} \qquad (1)$$

Continuing with the operations specified in rule 3, the acquired information is processed. Since no value exists for MARY, rules 5, 6, and 7 are applied and MARY is set up as a subgoal. A subgoal is treated as a goal and rule 3 (SOLVE) is thus applied to MARY. Proceeding as before, the ACQUIRE-INFO step ultimately yields

$$\text{MARY} = 6 + \text{FRED} \qquad (2)$$

The subgoal FRED is then processed leading to

$$\text{FRED} = 2 \qquad (3)$$

The MANIPULATE option of rule 5 can then be applied to equation 3 which recognizes that it has a value for FRED.

Lower level rules such as SUBSTITUTION and normal arithmetic operations (not shown here) can then be applied yielding solutions to the higher level subproblems previously generated, eventually leading to a solution of the original problem.

This problem illustrates one of the major features of means-ends analysis, namely, the creation and solutions of subproblems as intermediate steps between the problem statement and the ultimate goal. The hierarchic structure (generally referred to as a tree) shown in Figure 2 graphically illustrates the processes described. Each node represents an operation incorporated into the grammar rules specified earlier. Arcs emanating from a node lead to the constituent operations spawned by the more general

---

* We use the names of individuals to stand for the number of peaches each one has. We advocate the use of meaningful mnemonics rather than the traditional single letter variable.

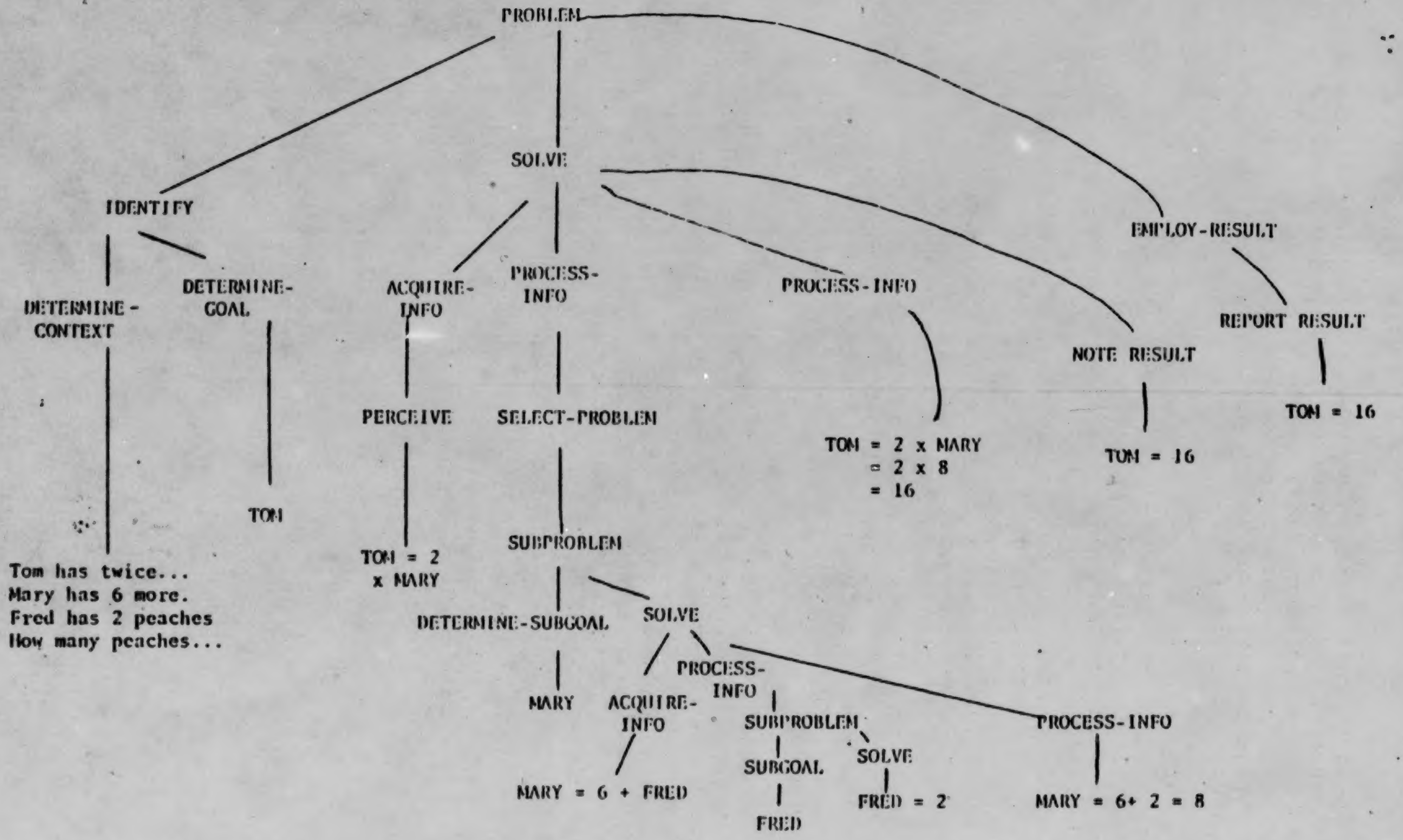

PROBLEM
IDENTIFY
SOLVE
EMPLOY-RESULT
DETERMINE-CONTEXT
DETERMINE-GOAL
ACQUIRE-INFO
PROCESS-INFO
PROCESS-INFO
NOTE RESULT
REPORT RESULT
Tom has twice...
Mary has 6 more.
Fred has 2 peaches
How many peaches...
TOM
PERCEIVE
TOM = 2 x MARY
SELECT-PROBLEM
SUBPROBLEM
DETERMINE-SUBGOAL
MARY
SOLVE
ACQUIRE-INFO
MARY = 6 + FRED
PROCESS-INFO
SUBPROBLEM
SUBGOAL
FRED
SOLVE
FRED = 2
PROCESS-INFO
MARY = 6+ 2 = 8
TOM = 2 x MARY
= 2 x 8
= 16
TOM = 16
TOM = 16

Figure 2

ones. The higher level rules shown, while general, still provide guidelines for analyzing the information contained in a problem statement sufficiently to establish the relationships necessary to solve that problem. The grammar rules define a variety of sequences of operations for accomplishing this.

The operative and descriptive subsystems are used to model an interactive problem solving situation, the operative subsystem emulates the student in the preceding example. As the production system acts on the problem, it produces a trace consisting of a list of actions taken. The list is input to the descriptive system which builds a partial description or parse of the sequence. The nature of a grammar limits the number of alternatives which can legally follow what has thus far been parsed. Therefore, the descriptive subsystem can provide "suggestions" to the production system if it cannot decide what to do next. Additions to the standard formalism of a grammar allow our system to associate with any node in its hierarchic description the goal under consideration and the context in which it is being worked upon.

Preliminary versions of the operative and descriptive subsystems have been implemented. Focus is on the domain of high school algebra word problems. The descriptive subsystem runs in ALISP (University of Massachussetts LISP) on the CYBER-73 at Oregon State. The operative subsystem is programmed in SNOBOL for the DEC-10 at the University of Oregon. The entire system will eventually be programmed in LISP. Since both systems are interpreted, no running times have been compiled.

Results, thus far have been encouraging. Hence we feel that the model can eventually become the basis of an intelligent, interactive system for problem solving instruction.

Bibliography

1. Brown, J. S. and Richard Burton.

   "SOPHIE - A Pragmatic Use of Artificial Intelligence in CAI". Proceedings of the Annual Conference of the Assoc. for Computing Machinery, Oct. 1974, pp. 571-579.

2. ----- and -----.

   "A Paradigmatic Example of an Artificially Intelligent Instructional System". Proceedings of the First International Conference on Applied General Systems Research: Recent Developments and Trends. Aug. 1977.

3. Ernst, G. and A. Newell.

   GPS: A Case Study in Generality and Problem Solving. Academic Press: New York, 1969.

4. Miller, M. A. and I. P. Goldstein.

   "SPADE: A Grammar Based Editor for Planning and Debugging Programs". MIT AI Laboratory Memo 386. December, 1976.

5. ----- and -----.

   "Parsing Protocols Using Problem Solving Grammars." MIT AI Memo 385. December, 1976.

6. Paige, J. M. and H. A. Simon.

   "Cognitive Processes in Solving Algebra Word Problems", in Kleinmuntz, Benjamin (ed). Problem Solving: Research Method, and Theory. Wiley and Sons: New York, 1966.

7. Polya, G.

   How To Solve It. Doubleday and Co.: Garden City, 1957.

8. Sacerdoti, E. D.

   "Planning in a Hierarchy of Abstraction Spaces", Artificial Intelligence 5, 2 (1974).

9. Wickelgren, W. A.

   How to Solve Problems. W. H. Freeman and Sons: San Francisco, 1974.

141

WHAT SHOULD THE FUTURE INSTRUCTIONAL THEORY FOR CBI LOOK LIKE?

Greg P. Kearsley
Courseware, Inc.

1. Introduction

Much is made of the fact that new technologies or innovative teaching techniques typically have no lasting effect on educational practice despite considerable promise. A major reason for this failure is that instructional theory does not keep pace with these technological advances, and hence, the potential of new technology is seldom utilized to any significant extent. This was true of television for a long time, is still true of computer based instruction (CBI), and will undoubtedly be true of videodisc systems. There is always a strong tendency to use new technology in the same ways as the existing technology. For example, how often have you seen an instructional television program which consists of 30 minutes of somebody lecturing in front of a blackboard? Can you imagine that technique being used in a television commercial (except as a joke)? Or what about the most unpardonable sin in CBI--automatic page turning--in which the computer does nothing more than present the screen-by-screen version of a textbook. The point is, that for every new technology, a new level of theory is needed to fully take advantage of new instructional capabilities. After all, there is nothing inherently wrong with the idea of presenting text on a CRT screen; however, there are many display and interactive capabilities (e.g., blinking, overlays, interspersed questions, individualized reading or difficulty levels, etc.) which should be used to make the instructional interaction more powerful and fully utilize the potential of the computer as an instructional medium or tool.

This article discusses the gap between instructional theory and the techniques of CBI. It begins by describing the nature of instructional theory, followed by a discussion of the theoretical models or frameworks associated with existing CBI systems. The final section outlines the future developments which CBI systems are likely to take and the corresponding instructional theory which will be needed. A major theme of the article is that research in artificial intelligence provides the basis for future CBI systems and instructional theory.

2. What Is Instructional Theory?

Instructional theory is concerned with factors which influence the effectiveness of instruction. In the most general sense, the goal of a theory of instruction is to prescribe the best methods for acquiring new knowledge and skills; in the narrowest view, it is concerned with optimizing (in a cost/benefit sense) a particular teaching sequence. Instructional theory encompasses the effects of learner attributes, incentives or motivation, the presentation and sequencing of curriculum, the measurement of student performance and instructional effectiveness, and the process of developing instructional materials. Although a full-fledged theory of instruction has not been proposed to date, many suggestions have been made (e.g., Atkinson, 1972; Bruner, 1967; DiVeste, 1972; Gropper, 1976).

A good case can be made for the proposition that CBI systems provide a highly suitable medium or methodology for the development of instructional theory, perhaps more so than any other approach within the realm of educational psychology. Ultimately, all ideas in CBI must become expressed with sufficient precision to be coded into programs which become the operational definition of the theory. This theory then undergoes testing in the form of student use under real learning conditions. Student response data is usually collected automatically, and the instructional effectiveness or consequences of the theory are empirically assessed. Thus, the instructional theory represented in CBI programs undergoes internal validity checks (consistent logic in the program) and external validity testing (through student response data). For these reasons, it is desirable that computers be used for research in all areas of education, e.g., counseling and guidance, testing and measurement, or teacher training.

If we confine our interest just to CBI systems, instructional theory can be focused on 3 major dimensions: the communication interface between student and system, the individualization methods employed, and the evaluation approach. Each of these three dimensions encompasses various parameters. The communication interface spans a number of input/output capabilities which may be grouped into three categories: verbal interaction (e.g., language or speech), visual interaction (e.g., graphics, color, motion), and tactile interaction (i.e., touching or pointing responses). The individualization methods can be categorized as performance-based (e.g., adaptive tailoring), personality-based (e.g., aptitude treatment interaction), student centered (e.g., learner control), and knowledge-based (e.g., student models, socratic dialogue). Since particular types of individualization methods embrace certain types of strategies or logics, the specification of an individualization approach is essentially the same as specifying a strategy. The evaluation dimensions can be described as either response-level (e.g., response analysis, feedback), lesson-level (e.g., item-generation, diagnostic tests), or curriculum level (e.g., comparative evaluations). Assessments of instructional effectiveness can occur at any level.

As Figure 1 depicts, the basic theoretical issues in CBI arise from the interaction of these three dimensions. The nature of student interaction will depend upon how they communicate with the system, the individualization approach, and the way the evaluation is designed. Each of these three dimensions is theoretically an independent domain; any type of communication interface could be combined with any type of individualization strategy or evaluation techniques. In practice, however, some combinations go together, such as the use of the natural language interfaces in socratic dialogue systems or the type of quantitative response analysis performed in the adaptive tailoring type of individualization. Each combination of dimensions suggests a possible theoretical approach to instruction. For example, one could envision a touch input/graphics output type of system with personality-based individualization and diagnostic testing for deaf instruction. Or the use of synthesized speech for feedback in a performance-based program to teach reading skills. Figure 1 provides a simple model to be used in classifying existing CBI systems and suggests some possible approaches to be explored.

What does this model have to do with the goal of instructional theory, namely to prescribe the best methods of designing and delivering instruction? The model suggests that the most effective computer based instruction will result when all three of the dimensions in the model had been simultaneously addressed. Instruction which focuses on one dimension without regard to the other two interacting dimensions will be inadequate and incomplete. Thus, complete attention to the individualization strategy in the design of a system, without equal attention to the communication interface and evaluation approach, will lead to instruction which cannot be totally effective. Furthermore, attention must be centered on the effects of specific interacting dimensions, since some of the combinations of variables will lead to more or less successful instructional outcomes. For example, the type of response analysis appropriate in a performance-based strategy (which would be quantitative) might be totally inappropriate for a knowledge-based strategy (which would be qualitative in nature). This point will be elaborated in the following sections.

3. Theoretical Frameworks Associated with CBI Systems

Every CBI system under development or in use today has an underlying theoretical framework which reflects the motivations or intent of the system designers. This section briefly surveys a number of existing systems in terms of the three dimensions introduced in the previous section. Since the systems surveyed represent some of the most advanced CBI efforts that exist, the state-of-the-art in instructional theory is reflected in this discussion.

It is most appropriate to begin with the PLATO system insofar as it was one of the earliest CBI systems developed and one which is quite widely used today. Although the instructional orientation of the PLATO system has always been avowedly eclectic, the majority of the courseware has been tutorial in style. The individualization strategies have followed the inclinations of the authors and have included all four types discussed previously. In terms of communications interface, PLATO allows the use of touch input and vector generated graphics. Response analysis in PLATO can be as sophisticated as is possible using syntactic (keyword-type) parsing. The CDC PLATO system includes a CMI capability which provides capabilities for diagnostic testing and test formatting.

The TICCIT system features a unique learner control type of instructional logic which makes the individualization approach student-centered. TICCIT was the first major CBI system to incorporate theoretical ideas on instructional display characteristics and sequencing into the system design and specification. The communications interface of the TICCIT terminal includes color graphics and video capability and the special learner control keyboard. Evaluation characteristics include the use of an "advisor" and "helps" to provide diagnostic capabilities at the lesson and curriculum levels.

The Stanford drill and practice programs in arithmetic and language arts (e.g., Suppes & Morningstar, 1972) and the work of Pask (1975) are representative of adaptive control approach to individualization. Considerable attention has been devoted to adapting the level of difficulty of the instruction based upon student performance. This approach tends to focus on lesson level evaluation with little attention to the response or curriculum levels. The communication interface tends to be very constrained, i.e., typed input/output.

Beyond the learner-controlled type of student-centered individualization represented by the TICCIT system are the "reactive" systems such as the LOGO projects (Statz, 1977), the SOLO project (Dryer, 1976), the Dynabook/SMALLTALK project of Xerox PARC (Goldberg and Kay, 1977), or the UC Irvine Physics curriculum (Bork, 1976). The common idea underlying these CBI projects has been to introduce students to the computer as a learning tool and encourage them to use it to explore some domain (e.g., mathematics, physics, etc.). This approach has resulted in some interesting developments in communication interfaces, such as Papert's "turtles", the simulation devices of the SOLO project, or the dynamic graphics in the Dynabook and UC Irvine's efforts. However, as a rule, these projects have provided enrichment or supplemental learning experiences, and as a consequence, little emphasis has been placed upon evaluation capabilities (at any level) in these systems.

A number of systems have been based upon a socratic dialogue type of instructional strategy. These include SCHOLAR (Collins, et al, 1975), SOPHIE (Brown et al, 1975), the BASIC Instructional Program (Barr, et al, 1976) or the FLOW Tutor (Gentner and Norman, 1977). These systems are characterized by a knowledge-based approach to individualization, since the adaptation to the student is based upon the content of what the student knows. Some very sophisticated work has been done on evaluation techniques in the form of error processes and "debugging" techniques (e.g., Brown, et al, 1977; Collins, 1976). Considerable emphasis has been placed within this framework on the development of verbal communication interfaces, particularly natural language and speech.

In addition to these major CBI efforts, there are numerous research developments which further extend CBI capabilities in the individualization, communication interfaces, or evaluation dimensions. For example, the Stanford MISS system (Sanders, et al, 1976) and the computer speech aids developed for the deaf (e.g., Nickerson, 1976) represent explorations of the use of speech in CBI systems. The CHARGE system developed by HumRRO (Swallow, 1977) is an exploration in three dimensional graphic displays. Work in the BASIC Instructional System on assessing "holes" in a curriculum network and automatic generation of missing content is an advance in the curriculum evaluation capabilities of CBI (Westcourt et al, 1977).

4. The Nature of Future Instructional Theory in CBI

As will be evident from the preceding section, each approach to CBI tends to focus on particular theoretical dimensions and pushes these forward. No single approach discussed above has covered the entire domain of theoretical dimensions underlying CBI; however, taken together, they comprise a theoretical framework which is close to doing so. To summarize in a sentence, the major trends in the development of the instructional theory associated with CBI are the development of a knowledge-based, student-centered typed of individualization, sophisticated verbal (i.e., speech and natural language) and visual (i.e., color, three dimensional, dynamic) communication interfaces, and the study of error-based diagnostic evaluation methods. The next generation of CBI systems which will ultimately emerge from this theoretical framework will likely involve a highly generative approach to all levels of evaluation, a dialogue-driven type of instructional process, and a communication interface which opens up rather than constrains the interaction possible between the student and the system. This differs from the instructional framework of current "production" systems (e.g., PLATO, TICCIT) in which the degree and nature of a student's interaction is limited to the pre-specified situation where questions are posed or tests occur. This does not allow a student to affect the course of learning by asking questions or to reveal subtle understandings or misunderstandings of the instruction through answers. In essence, current systems cannot provide a sophisticated level of individualization since they lack the advanced communication interface and evaluation techniques needed to achieve this.

If this point seems a little obscure, consider the nature of the individualization strategies, communication interface, and evaluation approach which exists with a good teacher and motivated student. A teacher will make adjustments to the level of the presentation (in terms of vocabulary, complexity of concepts, inferences required, etc.) depending upon subtle clues in the student's behavior, e.g., hesitation in responding or puzzled facial expressions. The teacher will model a skill or procedure to be learned, as well as the student's attempt, in order to indicate exactly where the student deviates from the desired behavior. The teacher will use communication techniques, such as voice intonation or hand motions, to emphasize an idea. A student will interrupt a presentation with questions to clarify a point or ask for further elaboration. The response to a student's question will depend not only upon the content of that question but also upon previous questions asked. These characteristics of teacher-student interaction reflect aspects in the instructional process which need to be included in sophisticated CBI systems.

The instructional theory required for the development of such advanced CBI systems encompasses questions as how visual, verbal, and tactile information is best related to different types of learning, how to generate problems or questions from subject matter on the basis of semantic operations (e.g., paraphrase, inference, reasoning), how to evaluate answers on the basis of semantic operations, how individuals seek information (i.e., knowledge acquisition strategies), and how to keep people motivated and mentally involved during learning. Many of these issues, and the more technical problems they include, are the subject of current research efforts in the domains of artificial intelligence and cognitive psychology (See Kearsley, 1978, for a survey).

5. Conclusion

Being able to fully capitalize on the potential capability of a medium reflects a good understanding of what these capabilities are. The knowledge of how best to use computers in education goes right along with an understanding of when not to use them. Thus, our present inability to use computers for their full instructional potential is the other side of our frequent misuse of computers in education. Research on advanced CBI systems is providing us with some idea of what these capabilities are. It is the role of instructional theory to synthesize these ideas and suggest guidelines for the design of future CBI systems. In particular, an instructional theory is needed for CBI which identifies important dimensions of individualization strategies, the communication interface, and evaluation approaches, and the effects of interactions among these dimensions. When this type of multivariate theoretical framework exists, it will be possible to prescribe instructional activities which genuinely require the power and sophistication of computer based instruction.

REFERENCES

Atkinson, R.C. Ingredients for a theory of instruction. *American Psychologist*, 1972, *27*, 921-931.

Barr, A., Beard M., & Atkinson, R.C. The computer as a tutorial laboratory: the Stanford BIP project. *International Journal of Man-Machine Studies*, 1976, *8*, 567-596.

Bork, A. Learning, computers and pictures. *THE Journal*, 1974, *1*, 6-13.

Brown, J.S., Burton, R.R., & Bell, A.G. SOPHIE: A step towards a reactive learning environment. *International Journal of Man-Machine Studies*, 1975, *1*, 312-349.

Brown, J.S., Burton, R.R., & Larkin, K.M. Representing and using procedural bugs for educational purposes. *Proceedings of ACM Convention*, Seattle, November, 1977.

Bruner, J.S. *Toward a theory of instruction*. Cambridge, Mass.: Belknap, 1967.

Collins, A., Warnock, E.H., & Passafiume, J.J. Analysis and synthesis of tutorial dialogs. In G. H. Bower (Ed.), *The psychology of learning and motivation*, *Vol. 9*. New York: Academic Press, 1975.

Collins, A. *Processes in acquiring knowledge*. Bolt, Beranek & Newman, Report #3231, January, 1976.

DiVesta, F.J. An evolving theory of instruction. *Educational Technology*, 1972, *12*, 34-39.

Dwyer, T.A. Heuristic strategies for using computers to enrich education. *International Journal of Man-Machine Studies*, 1971, *3*, 219-239.

Gentner, D.R.& Norman, D.A. *The FLOW tutor: schemes for tutoring*. CHIP #70, University of California, San Diego, June, 1977.

Gropper, G.L. What should a theory of instruction concern itself with? *Educational Technology*, 1976, 7, 7-12.

Goldberg, A., & Kay, A. Teaching SMALLTALK. Xero Palo Alto Research Center, SSL 77-2, June, 1977.

Kearsley, G. P. The relevance of AI research to CAI. Journal of Educational Technology Systems, 1978, 6 (3), 229-250.

Nickerson, R.S., Kalikow, D.N., & Stevens, K.N. Computer-aided speech training for the deaf. Journal of Speech and Hearing Disorders, 1976, 41, 120-132.

Pask, G. Conversation, cognition and learning. New York: Elsevier, 1975.

Sanders, W., Benbassat, G. & Smith, R.L. Speech synthesis for CAI: The MISS system and its applications. ACM SIGCSE Bulletin, 1976, 8, 200-211.

Statz, J. Current work and plans with LOGO. ACM SIGCUE Bulletin, 1977, 11, 3-7.

Suppes, P., & Morningstar, M. Computer assisted instruction at Stanford, 1966-68. New York: Academic Press, 1972.

Swallow, R. J. CHARGE and its potential in CAI. In R. J. Seidel & M. Rubin (Eds.), Computers and communication. New York: Academic Press, 1977.

Westcourt, K.T.,Beard, M., & Gould, L. Knowledge-based adaptive curriculum sequencing for CAI: Application of a network representation. Proceedings of ACM Convention, Seattle, November, 1977.

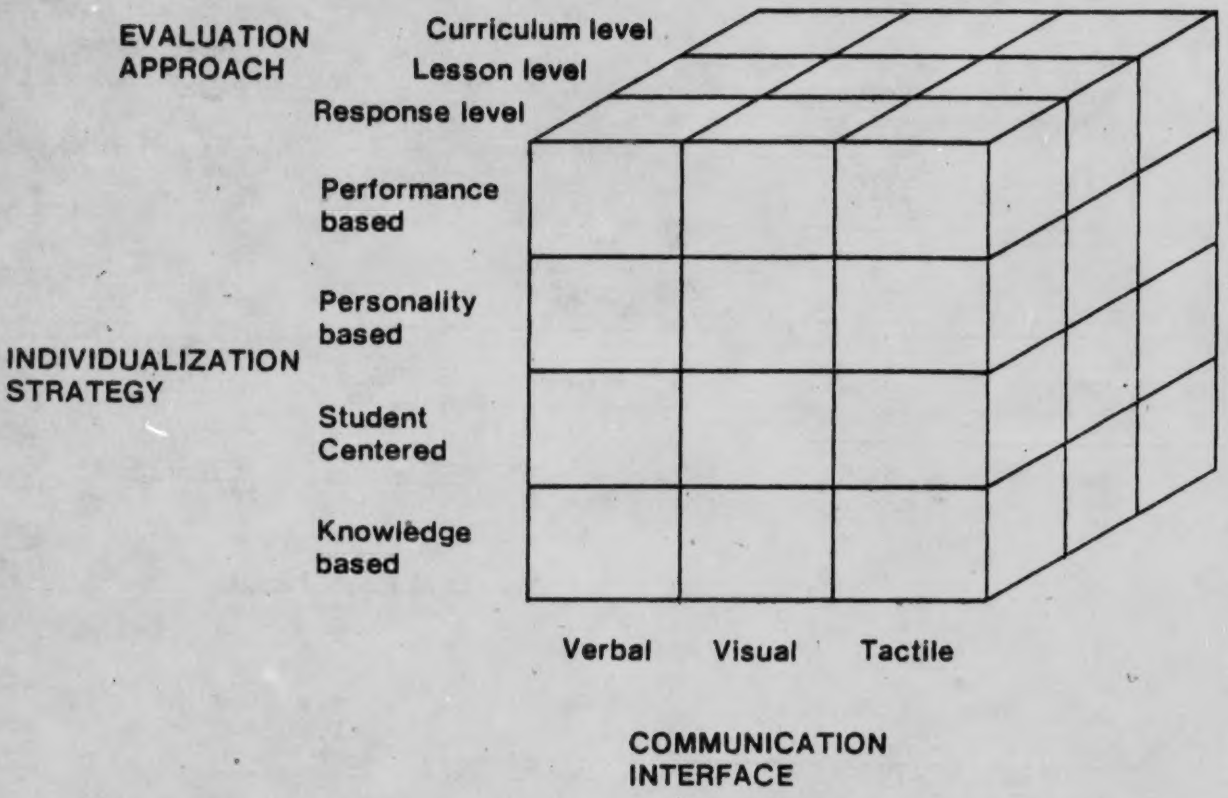

EVALUATION APPROACH
Curriculum level
Lesson level
Response level
Performance based
Personality based
INDIVIDUALIZATION STRATEGY
Student Centered
Knowledge based
Verbal
Visual
Tactile
COMMUNICATION INTERFACE

Figure 1. Instructional dimensions in CBI.

Aldwyn J. R. Cooper
The Open University
Milton Keynes, England

A preliminary report on the development and application of CICERO, an interactive and batch computer assisted/managed learning system developed for use in the distance teaching environment.

The Open University is a distance teaching establishment. It runs degree and post-experience courses in a wide range of subject areas. Its student population is currently in the region of 60,000, the majority of whom have had some break in their education prior to embarking on a degree course. Entry is 'open' in that there are no formal academic requirements and applications are handled on a 'first-come, first-served' basis. The central academic campus of the university is at Walton Hall, Milton Keynes. In order to adequately service a student body distributed throughout Great Britain, thirteen,regional,satellite administrative centres are maintained. In addition to these there some 200 study-centres where students occasionally attend for limited face-to-face tuition. However, it has always been the aim of the university to provide 'distance education' where, while face-to-face tuition may be important, it is not a preeminent factor in the teaching program. The Main Elements of all courses are transmitted by correspondence texts, set books, radio and television broadcasts and home-study kits.

Courses are designed, written and assembled by teams of academics at the Walton Hall campus. In addition to writing course material the team is required to construct an assessment profile. Traditionally this has been a blend of continuous assessment and examination and has been comprised of four elements. The first is Self-Assessed-Questions (SAQs) or In-Text-Questions (ITQs). These are for the students' benefit alone. Answers are provided in the same set of correspondence material. The option is clearly there for the questions to be ignored entirely or for the student not to exercise him/herself adequately before reading the answer. The second, examinations, is clearly for the benefit of the university alone. Except as a possible salutory lesson or reinforcer for the future, there is no educational feedback value in an end-of-course examination. The third is Tutor Marked Assignments (TMAs). These take the form of traditional essays which are sent to tutors for grading. It is returned marked and annotated to the student and a grade is retained by the university for use in the final degree conflation. This has generally proved to be the most satisfactory of the assessment factors. However, the method is expensive in tutor time, suffers from the problems of variable and non-standard marking and may involve delay for the student.

The final element is the Computer Marked Assignment (CMA). This has normally taken the form of a multiple-choice test with up to forty items. Students are required to complete one of these on each 'Block' of a course. The form is then sent into the Data Processing dividion of the university prior to a specified cutoff date. All the forms are optically read and machine marked. A grade is obtained to be added into the degree conflation and the student is informed. This grade letter is the only feedback that is sent to the student.

During the seven years operation of CMAs there has been a growing dissatisfaction with their utility on all counts. If they are to be used for assessment, security dictates that a new set of questions be written each year. Even ignoring the argument concerning the limited reliability of multiple-choice questions generally, it is a difficult and time-consuming task to produce good, plausible questions. Once a course has been written the course team splits up and the maintenance role of CMA production devolves onto one person who may not have been involved with the original writing. Despite the possibilities offered by item banking overall this is a time-wasting and academically dubious method of practice.

Such sets of multiple-choice questions delivered in this necessarily unsupervised manner are obviously susceptible to collaboration. Indeed, it has been suggested that this is their only beneficial characteristic. The desire to achieve better grades is the magnet which draws the students together to discuss course work. However, it is questionable whether the consensus of ten or twenty students working together should be included in any individual student's grading.

Finally, there is no pedagogical advantage to be gained from this method. The constraint of the cutoff date means that some students will not receive any feedback for up to six weeks by which time they will have progressed to other areas of the course. Even should they have time to return to their earlier work, the provision of a single grade letter does not permit the possibility of identifying their particular problems. There has been a growing dissatisfaction and an increasing belief that the potential for the use of computers in education could be better realised within the university.

The Student Computing Service (SCS) and CICERO.

The Student Computing Service is a wholly separate unit and has no Data Processing responsibilities within the university. It was originally founded to provide computing resources for students on mathematics courses. Since then its scope has widened and SCS now provides a wide range of facilities for maths, science, technology and social science faculties.

It operates from three regional computer centres, one at the central campus and the others at London and Newcastle. Each centre is equipped with a Hewlett Packard 2000F machine and a Digital DECSystem 20/40. All development and research work is carried out at the central unit which also has Access links to IBM 370 and Univac systems. Students access the machines through a network of over 200 terminals based at the study centres. For those students who, for whatever reason, find it impossible to attend a study centre a postal batch system is maintained wherever this is feasible.

In September 1976 SCS set up a research team to develop a system which would provide rapid feedback to students on their progress through the course, could provide some tuition and remediation interactively at the terminals and could provide direction for future study. It was hoped that the system could be an eventual non-assessed, formative replacement for the CMA system and could also be viewed as a powerful supplement to the limited tutorial provision. Part of the philosophy of the research team was to field test a pilot system as quickly as possible and it was planned to commence operation in February 1977.

This system was designed to operate as follows. A set of carefully constructed 'profile questions' would be sent to students at the end of each block of the course. These would not be for summative assessment of the students. Rather, each question would be constructed such as to provide diagnostic information concerning strengths and lacunae of knowlege about specific objectives of the course. Any question could score to any combination of these objectives. The scores achieved in the profile questions were termed level one scores.

Once a student had completed answering the profile questions he/she had two options on how to progress. If it was quite impossible to attend the study centre, the answers were posted off to SCS. They would be coded into the machine on the day of receipt, processed over night and returned by the next post. The output to the student took the form of a series of annotated answers to the questions based on the student's individual responses. The CICERO program also inspected the patterns of level one scores and provided advice on courses of study. The postal system alone provided a considerable improvement on the old CMA system.

The second, and more beneficial option open was to take the completed answer form to the study centre and call up the CICERO program direct. In the interactive mode the program prompted for the student's responses to the profile questions and then provided annotated answers immediately. The level one scores were then interpreted. If some doubt existed about the student's competence with respect to a particular course objective, additional questions could be asked and answered on-line. These resulted in level-two scores. If level-one alone, or the combination of levels one and two indicated that a particular objective was causing difficulty the student could be led through a remedial package on the machine. Scores on remedial packages accrued to level-three. If no problems were experienced, or the necessary remedial tuition had been successfully completed the student could be led on to advanced problems. These scored to level-four. At the completion of all the work the program examined the pattern of scores for all the objectives at all four levels and provided advice contingent on these. A record was retained on the system of every student attempt on any tutorial and all the scores that were achieved. These could be used for later assessment and rewriting of the computer tutorial and/or the course itself.

This first pilot system was written in Hewlett-Packard Extended Basic and was run on the 2000F machines. A specialised prompting editor (CEDIT) was written to set up the tutorial files. These were constructed as a series of sequentially numbered nodes. A node being defined as a point at which a processing operation occurred. When used interactively this file was interpreted by the main control program CICERO and when used batch was interpreted by the postal control CICERP. Two further programs were developed also. One specially formatted postal input data (CICP5) and the other printed out scoring files for inspection (CICP3).

During 1977 the system was run with c.2000 students on three courses in Psychology,Education and Economics. Approaching 70% of students used the system in one of the forms (it was add-on and optional). Approximately two-thirds of these used the interactive version. These figures compare favourably with the 40% of students who regularly attend the face-to-face tutorials and perhaps reflect the benefits accruing from flexible hours of use. In addition it had a demonstrable appeal to academics in terms of ease of maintenance, feedback, easy susceptibility to change and the fact that, because it is not for summative assessment, once produced the major part of the material lasts the life of the course.

CICERO-2

Considerable experience was gained during the first year. Problems of inflexibility were identified. Innefficiencies were detected in the specialised editor. Much student and staff comment was reviewed. At the same time, increased demand for SCS services had led to the purchase of the three DEC 20/40 machines. The most effective course of action appeared to be the complete rewriting of the system in DEC Basic +2.

The new system was written to supply a more flexible and user-friendly approach for both the author and the student. An 'Author Language' was developed to replace the cumbersome prompting editor. In the new system authors structure their tutorials in the author language using the normal DEC-20 editor. This allows material to be input and corrected considerably faster than by the old method. Once an author is satisfied that the tutorial structure is correct the file is translated into control-file format by use of a new program (CATO). This program provides a cross-referenced listing of the tutorial file and a control file. All viable, translated tutorial files are added to a course index.

A new CICERO program has been written to accept the new type of control files and to access them through the index. A student entering the system provides the necessary accounting information and a unique student number which acts as a key to all records and permits restarting of tutorial runs. Once this number has been validated the student can peruse the index or immediately specify a tutorial. A message facility has been incorporated at this point which allows tutors or administrators to address students through the system. This facility can be focussed down onto a single student, can apply to sub-groups of the population or the whole population.

In the old system it was essential that the student had received and completed the profile questions before attending at the terminal. While this is still desirable in the new system in order to save time, it is now possible to bring up the full text of the profile questions on-line. This is achieved by use of one of the commands in the 'Student Command Language'. This is another new feature of the revised system which allows the student far greater control over his/her own passage through the material. There are commands which allow the student to transmit copies of the terminal run by post to a tutor, which allow the use of the system for electronic mail, which allow the skipping of all or part of remedial packages and a number of others.

The new system also keeps far more comprehensive records than the old which allow the complete reconstruction of the path taken by every student and note every response that the student made. These records are maintained in an 'Overnight-Index' with a key to their status, complete or non-complete. The overnight-index is tripped of all complete files on a regular basis and these records are compressed and appended to a 'Student-Main-File' where all the information on system useage is kept. A series of programs have been,and are being, developed which allow interrogation of the student-main-file in a variety of ways providing information as specific as how an individual student answered a particular question to mean performance of the entire population. These records should prove of considerable assistance to tutors and course authors.

The new system has been in operation since February 1978 and complete data on the year's run is not yet available. It has been available to approximately 10,000 students in a variety of ways.It has been used as:a straight diagnostic-tool to assist in developing mathematical competence for students on a science foundation course, a straight tutorial course in education, psychology and economics, and as a research tool mapping the ways in which primary school children learn.

Precise figures and reactions will be published as soon as they are available. The demand for the system is increasing all the time and many developments are envisaged. The most exciting of the new possibilities is the use of the system in the student's own home either by means of low-cost terminals, the British Post Office Prestel system or by micro-processor systems.

"How Should We Train Teachers and Administrators in Computer Application to Education?"

DR. MICHAEL R. MOCCIOLA

COMPUTING SYSTEMS MANAGER
AND LECTURER

SCHOOL OF EDUCATION

WESTERN AUSTRALIAN INSTITUTE
OF TECHNOLOGY

PERTH, AUSTRALIA

Introduction

The purpose of this paper is to describe a teacher and administrative training technique used in pre-service and in-service courses in instructional and administrative computing for elementary, secondary teachers, and school administrators, at the School of Education of the Western Australian Institute of Technology (WAIT). Before describing the training technique central to this paper, a brief sketch regarding educational computing and teacher training today, may be useful.

Recent findings from both American and Australian computer educators suggest that instructional and administrative computing is increasing in the elementary and secondary schools of both countries, albeit at a precariously slow rate. However, courses in instructional and administrative applications of computers to schools in teacher training institutions, with a few notable exceptions, are virtually nonexistent. This state of affairs exists despite the heroic efforts of many educators, computer educators, national and international conferences, and journal articles attesting to the importance of computing within teacher training programmes. Additional evidence of this deficiency in educational computer training for educators is dramatically brought home by the realization that there is not one school of education in the United States or Australia today which offers a degree in computer education, though this is apparently to change in the near future. And, as might be guessed, there are precious few schools of education today which possess their own computing systems inspite of sharply decreasing computer costs coupled with miniaturised technology.

That there exists a clear need for well planned and organized educational computing courses,and programmes for teacher and administrative training is easily documented. However, what is not so clear is given that a school of education does or wishes to participate in instructional and administrative computing and can develop suitable courses in terms of syllabi, what is the best means to transmit the course content to students, teachers, and administrators.

Purpose of the Paper

It is the specific purpose of this paper to address this question of how to provide training within courses in computing for educators. In particular, this paper describes, illustrates, evaluates, and gives a rational for student group computing projects to be employed as a learning technique within teacher training courses in the applications of computing to education. It is argued that group computing projects are at present a necessary component in teacher training courses in educational computing. A series of propositions are stated to support this argument; the conditions and arrangements for assessment and administration of student projects are discussed; one example of such projects eminating from each of three different computer education courses at the School of Education at WAIT are given, and an evaluation of the student computer education projects is made.

Courses and Computer Facilities within the the School of Education

So that the reader does not suggest that the author is putting the proverbial "cart" before the "horse," a description of the computing facilities within the School of Education at WAIT and our educational computing programme may first be in order.

The School of Education (WAIT) is located about 3 miles from the Indian Ocean, just outside the city of Perth in Western Australia. The Institute has approximately 11,000 students of which 1,000 are enrolled with the School of Education in undergraduate and graduate programmes.

The School of Education has the following computer facilities: (it may be of interest to note that the hardware employed is manufactured in the USA)

1. A Digital PDP 11/34 computer with 64K words of memory.
2. Sixteen terminal stations (half of which are visual display units, half DEC writers).
3. Tektronic graphics.
4. A card reader and line printer.
5. Two television monitors.
6. Two computer accessed slide projectors.
7. Sixteen terminal access to the Central Computer Centre's DEC 10.
8. A small computer education library containing both books and relevant journals.

Courses in Instructional and Administrative Computing

The School of Education (WAIT) conducts three courses in educational computing for pre-service and in-service teachers, and school administrators. The general purpose of the courses is to allow students and educators to:

1. acquire the historical, philosophical, and psychological concommitants necessary for purposeful school computing;
2. review and discuss relevant research findings from various areas in education, such as curricula development and strategies of teaching, which can assist in the successful employment of computers in schools;
3. acquire a through-out-the-school perspective on educational applications to computing.
4. acquire the software and general hardware background necessary to assist in the initiation, implementation and development of a school's computing programme.

The School of Education offers two one semester elective courses (without prerequisite) to pre-service and in-service teachers, and administrators, namely: Computer Assisted Learning, Computers in Learning. Also, Instructional Programming for Teachers I and II is offered over two semesters.

Computer Assisted Learning introduces students to the author-languages COPILOT and PILOT, incorporates theories of computer assisted learning, and gives students the opportunity to create their own computer programmes in the author-languages mentioned. Computers in Learning discusses various modes of computer applications to learning, such as drill and practice, tutorial, dialogue; the possibilities and limitations of computers to education are discussed, historical, philosophical and psychological developments and points of view are entertained. Artificial intelligence and student controlled programmes are discussed. Instructional Programming for Teachers introduces students to the BASIC-PLUS language, demonstrates the use of a large number of packaged computer materials applicable to teaching, and provides opportunity for students to write their own programmes using BASIC-PLUS, relative to educational theory.

The School of Education does not offer a degree in computing education.

A Reason for Group Projects in Teacher Training Courses in Educational Computing

A number of propositions are to be made about the way successful computing applications are done or can be done in schools based on the evidence acquired by this author as manager of the School of Education's Computing System, as computer coordinator for a school district in New Jersey, and as a school computing consultant to public and private schools in Western Australia. If these propositions are accepted as valid then a strong case exists to provide a training technique which employs significant group computer activities in teacher and administrative computing applications to education.

The Propositions

1. Usually, students, teachers, and administrators interact within groups when addressing significant educational issues; and it is this group interaction that may be a major factor in the success or failure of educational programmes within a school district, particularly if that programme has a computer component.

2. Educational projects which are computer related more often than not cross a number of discipline/ administrative areas. This relation inevitably means the interaction of educators for each area represented in the project.

3. In schools, it is relatively rare to find educational software development and computer programming being done by a single teacher. Most school project developments (where projects are being developed) are completed by a small team of teacher and administrative specialists.

4. Computer-education is a complex issue involving not only mastery of various programming skills, but also knowledge of human learning, classroom management, school organizational structure, etc. Such knowledge and its application to computing in schools seems best to be acquired and transmitted through groups.

5. Computer education specialists are rare and likely to be so for the immediate future. This dearth of specialists usually necessitates the incorporation of other staff members into a computing school project for it to be successfully completed.

6. An underlying assumption to successful school computing is that thinking of the computer's role in educational applications means first thinking about education, not computers.

General Features of Group Computer Projects

If the reader is willing to accept most or all of the above propositions, then group computing projects for computing education training may appear quite reasonable. We now take up the question of "What kind of group projects best satisfy the computing requirements of educators and schools, and, how can they effectively be placed in computer education courses?" The approach that has generally been adopted in our computer education courses at WAIT is that the students solve problems within the areas of school administration and instruction with the aid of the computer through mutual participation, evaluation and correction. It is important to reinforce that the problems given to groups of students to solve are educational problems first, not computer problems.

In order to satisfy the objectives of the courses the projects are simulations of the educational needs of the school. Such simulations prepare students directly for school situations as well as enable them to acquire the learning objectives of the computing education courses. A good simulation is also good for morale and performance, as the participants can see the relevance and benefits of the exercise.

It is desirable that the projects be such that groups have to make decisions about the project goals. Although projects must be designed to be limited in scope, groups should be able to make choices between various goals as well as decide their own internal group structure and organisation.

Among the most important aspects in the success of projects is staff commitment. Group projects require a considerable amount of effort on the part of the instructor to make them worthwhile and realistic. This is also important since group projects are often a unique experience in a student's education and therefore is wary of them.

Projects do not exist in isolation from workshop/ lecture activities. In our case workshops and lectures are arranged to support the various stages of the projects.

For efficiency of learning we have found that student group sizes of 3-4 are most beneficial. If less, then the communication activities and relational values seem to be negligible. If more, the problems the groups encounter in trying to arrange logistics such as when and where to meet, start to outweigh the learning activities.

Assessment of Computer Projects

In assessment of student educational computing projects of the courses, the educational solutions are most important and are assessed first. For example, we are first concerned with the assessment of the following:

are the psychological and cognitive learning components satisfied;

has the educational problem been adequately addressed;

how do the students, or staff, which use the results of the project actually benefit;

does the result of the project cause other serious constraints to the students, staff, school, or classrooms that would be involved in using the programme;

is the completed project manageable within the existing structure of the school;

is the project generalizable to other structures with the school, such as other grade levels or other administrative uses.

Secondly, the technical aspects are assessed. For example,

are the computer programmes well documented (internally and externally);

are they transportable;

are they relatively efficient in design;

can files be adequately developed and maintained;

are the files secure from unauthorized users.

A Sample of Group Educational Computing Projects

1. This project is taken from the course *Computer Assisted Learning*.

   You have been advised by your principal that reading scores in grade 5 are generally below national averages. An investigation of the causes of these poor test results by students has several causes, one of which seems to be some students not having enough practice with various types of "comprehension-paragraph" assignments. You are to provide a drill and practice computer programme in reading suitable for your school's 5th grade reading classes so as to assist in increasing their comprehension skills.

   Your students are of average ability as a group, coming from middle class families in a suburb of Perth.

The textbooks generally used by the 5th grade are from the SRA series.

The reader may feel that this project is not well defined, e.g., more information about the students is needed. Some projects are purposefully kept "open" to encourage "real school world" application. Others, must be idealized in part by the restrictions of the college year. Students in Computer Assisted Learning write programs in the author language COPILOT or PILOT.

2. This project is taken from the course Computers in Learning

   The principal of your high school has collected data on students which indicate that many first year algebra students miss a significant number of meetings of their algebra class due to various factors such as illness, school plays, sports meetings, guidance meetings, tardiness, etc. In addition, these students frequently have difficulty in making up missed material. The principal would like you to develop a tutorial-computer-assisted-learning programme to assist these students in various areas of their algebra deficiencies. Choose a topic likely to cause trouble with the first year algebra students and design, write, and test a programme to provide tutorial assistance.

Due to the limit of the amount of programming skills of students in the course, topics are limited by the instructor. Emphasis is placed on the educational rational needed for development of the computer programme. Students develop programmes in an author language or BASIC-PLUS.

3. This project is taken from the course Instructional Computing for Teachers.

   The Child Study Team within your high school needs to be able to classify some students along various cognitive, psychological, and sociological domains to insure for remedial student needs. Though the Team has collected a mass of data, no satisfactory non-machine filing system of students has been developed.

> You have been asked to create a more efficient and effective computer filing system that will store and sort information about students. The system should be able to accept 300 names of students, their addresses, phone numbers, grade levels, one score on each test related to cognitive, academic, and sociological measures. Student information should be accessible to the Child Study Team by various sorts, or keys; such as student's last name, grade level, and minimum test levels. The file should be random access, easy to manipulate, and allow for addition and deletion of student information.

This course has a heavy programming orientation and students apply the language BASIC-PLUS.

Evaluation of Educational Group Computer Projects

Over 100 students have, in the past three years, completed group computer projects aimed at education or school applications. Our evaluation of the projects, to date, have centred around their usefulness in real school situations.

Students who have entered the school world have been asked to report back to us on the appropriateness of their learning exercises, the appropriateness of the topics, and the use to which they are implementing their computer education training. Their replies are anecdotal and descriptive, and therefore so is our evaluation to date. Though it is the case that a good proportion of our students have never used their training due to lack of computing equipment within schools, those who have, report generally highly favourable responses to our queries. For example, previous students found the group interaction particularly useful in that "this is what we really do" in schools. Also the educational prerequisites necessary to successful programme writing which have been thoroughly discussed in our WAIT courses, were found particularly useful when attempting to explain the needs and benefits to school personnel of a proposed computer project. It was found in a minority of cases that the actual programmes written as group projects were at least, in part, transportable to a school site in both a physical or application sense. This was expected inpart, since most student programmes were written in BASIC-PLUS, while most school equipment uses a different version of BASIC. In the application sense, rare was the case in which a student computer project fitted in a one-to-one correspondence with a particular school need. However, the group assignments to students were found to closely approximate real school needs.

Based on our first evaluative efforts, we believe our programme is sound, at least from the perspective providing appropriate computer education training for school applications. Some adjustment may be necessary, for example, we may wish to replace COPILOT and PILOT with a single and more powerful author-language such as DECAL-II. It may be more effective for our students to get group computer projects directly from schools rather than from a file of projects that instructors have suggested. This school liaison might be productive in more than one way in promoting successful computing applications to education. The language BASIC is being standardized, and this may help in conversion problems.

It may be useful to close this paper indicating that our experience at WAIT seems to corroborate ALLAN ELLIS'S remark, that thinking about the computer's role in education does not mean thinking about computers. It means thinking about education.

Toward a Computer Coach for Elementary Programming

Mark L. Miller[1]

Texas Instruments

Abstract

How could an appropriately structured environment facilitate the acquisition of programming skills? Significant theoretical strides are needed before human-quality performance can be expected from a computer-based programming tutor. As an intermediate step, a system has been implemented which serves primarily as an editing language and diligent clerk. However, it differs from conventional programming environments in two crucial ways: (1) it interacts with the student using a vocabulary of concepts about planning and debugging, derived from an explicit model of the design process; and (2) it actively prompts the student with a menu of design alternatives, within the overall framework of a mixed-initiative dialogue. The current system is not a tutor; but the process of implementing and testing it has been instrumental in refining our model of the design process, thereby bringing us a step closer to realizing a computer-based programming tutor.

Introduction

Within five years, every American who owns a television set will be able to afford a powerful general purpose digital computer. This situation presents an incredible opportunity for enhancing the quality of life in general, and of education in particular. It would be a tragic waste if this vast resource were squandered on trivial amusements such as "pong," or restricted to useful but mundane applications such as balancing the checkbook.

One of the more imaginative possibilities is the introduction of computer programming as a subject in the elementary school curriculum. A growing community of researchers argue that programming should be taught, not only for its intrisinc importance, but also because it can serve -- just as plane geometry has served -- as a vehicle for imparting problem solving skills.

Unfortunately, very little is known about the process of writing programs, and even less is known about how to teach it. For example, the astute employer would rather hire a skilled Algol programmer -- with no background in Fortran -- to write Fortran programs, than a new graduate trained specifically in Fortran. Why? That many languages have similar constructs, and that the veteran will know what to look for in converting to the new language, are only partial answers. What is it that the proficient programmer knows, that the beginner does not? The neophyte lacks knowledge of style, of strategy, of how to organize work on a large project, of how programs evolve, and of how to track down bugs.

This paper reports on an investigation of this knowledge, and describes a model which formalizes it as sets of rules. In accord with the above intuitions, only the lowest level rules in the model deal with the constructs of particular programming languages. The most important rules deal with plans (which are independent of the detailed form of the code), debugging techniques, solution-order (search) strategies for choosing what to work on next, and methods for exploring interactions among design alternatives.

A Vehicle for Studying Programming Knowledge

As a vehicle for pursuing this investigation, we have implemented a computer environment called Spade-0 which interacts with students who are learning to plan and debug elementary programs. Spade-0 prompts the student through an hierarchical planning process, encouraging the student to postpone premature commitment to the detailed form of the code. It explores the hypothesis that articulating one's problem-solving strategies facilitates learning, by providing a vocabulary of concepts for describing plans, bugs, and debugging techniques. It handles the routine bookkeeping tasks, and understands commands which edit the plan -- rather than the code -- of the student's program.

Our long term research objective is to construct "artificially intelligent" learning environments capable of human-like tutoring. Our approach to this difficult goal is to construct less ambitious limited didactic systems as subgoals. These intermediate systems are tutorial in nature, but limited in scope. They do not display the insight or sensitivity which we ultimately desire, but they help us to explore various facets of cognitive and pedagogical theories. Spade-0 in particular has been specifically designed to support systematic variation of its configuration of features, an essential property both for validating the environment as a whole, and for assessing the relative contributions of each module.

Turtle Graphics Serves as an Example Programming Task

Although our investigation has touched on several problem domains, our most extenive efforts have fallen within the realm of Papert's [1971] "Logo Turtle" language for simple graphics programming. Turtle Graphics has been a helpful source of examples because: (1) the complexity of typical Logo programs is manageable, even though the potential difficulty ranges over a wide spectrum; and (2) we have ready access to considerable student performance data for introductory Logo sessions.

The minimal Logo turtle language consists of these commands:

FORWARD ⟨number⟩	;Turtle moves ⟨number⟩ steps in current direction;
BACK ⟨number⟩	;Turtle moves ⟨number⟩ steps in opposite direction;
RIGHT ⟨number⟩	;Turtle turns ⟨number⟩ degrees clockwise;
LEFT ⟨number⟩	;Turtle turns ⟨number⟩ degrees counterclockwise;
PENDOWN	;Subsequent FORWARDs, BACKs draw lines;
PENUP	;Subsequent FORWARDs, BACKs do not draw lines.

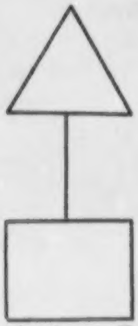
The user extends the available set of commands by defining new procedures. Each new procedure may be defined to accept zero or more formal parameters; each line of the new procedure is either a call to one of the Logo primitives, or an invocation (with actual parameters) of some user-defined Logo procedure. Conditional statements (IF -- THEN -- ELSE) and looping constructs (WHILE -- DO) are also available.

Figure 1 shows a Logo procedure and the "Wishingwell" picture which it draws; this serves as our example task. (WELL and ROOF are user procedures defined in the same fashion.)

Figure 1.
Wishingwell: A Typical Introductory Logo Project

```
TO  WW
10  RIGHT 90
20  WELL
30  FORWARD 50
40  LEFT 90
50  FORWARD 100
60  LEFT 90
70  FORWARD 50
80  RIGHT 120
90  ROOF
END
```

(a) Logo Code



(b) Resulting Picture

A number of students (generally high school) have been asked to solve this task. Our usual procedure is to ask them to first describe the picture in English, then to draw it by hand. Next we ask them to think aloud as they design the program using the Logo system. We collect a verbal prototcol as well as the typewritten transcript of the session. Except for the omission of typographical errors, the solution shown in figure 2 is typical.

Figure 2. A Typical Student Solution

```
E001   ? TO WW
E002   > 10 TREE
E003   > 20 WELL
E004   > END

E005   ? TO WELL
E006   > 10 REPEAT 4 [20 30]
E007   > 20 FORWARD 100
E008   > 30 RIGHT 90
E009   > END

E010   ? RUN WW
E011   ? CLEARSCREEN
E012   ? RUN TREE
E013   ? RUN WELL

E014   ? EDIT WW
E015   > 15 LEFT 90
E016   > 17 BACK 50
E017   > END
```

Modelling Elementary Programming

Spade-0 is based on an information processing model which accounts for this type of protocol.[2] While a complete description of this model is beyond the scope of the present paper, this section provides sufficient background information to explain the operation of Spade-0 and to argue that its implementation has been instrumental in suggesting refinements to the model.

The first thing which we notice about the protocol is that the problem-solving behavior can be partitioned into distinct phases or "episodes" [Polya 1965]. Even before the typescript begins, there is a phase of understanding the problem, that is, of converting the pictorial representation into some internal symbolic representation. The model does not address this: it requires as input a symbolic representation of the specifications for the desired picture (ala [Goldstein 1974]) such as is shown in figure 3. (The specification language is based on common-sense predicates such as "is-a" and "is-part-of", and geometry-specific relations such as "is-above" and "is-connected-to".)

Figure 3. Specifications for Wishingwell

Parts of WW: Roof, Pole, Well

Properties of Roof: Triangle, Horizontal

Properties of Pole: Line, Vertical

Properties of Well: Square, Horizontal

Relations Between Roof and Pole: Above, T-Connected

Relations Between Pole and Well: Above, T-Connected

After understanding the problem, there is a phase of defining a procedure (E001 - E008). The model claims that the definitional phase is the externally observable manifestation of an internal process of Plan Refinement. Other phases involve trying out the procedure (E010), localizing an error or "bug" (E011 - E013), and repairing a bug (E014 - E017). A separate "mode" is used to model each such phase: a mode is a packet of rules with its own interpreter. For example, Localization Mode is a set of rules which become active when Tryout Mode has discovered a bug; its job is to decide which part of the program is responsible for the error.

"Control Mode" is responsible for interrupting the normal flow of operations to switch modes, when appropriate. Control Mode includes rules such as:

a. If in Tryout Mode and an error occurs, enter Localization Mode with a description of the error;

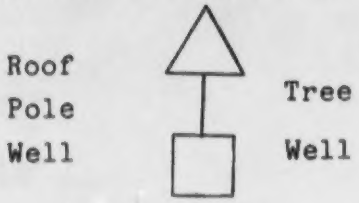
b. If in Refinement Mode and the definition of a subprocedure is completed, enter Tryout Mode for that subprocedure; and

c. If in Repair Mode and the underlying cause of the bug was found to be a missing step in the plan, enter Refinement Mode with the specification for that step as goal.

Since all of the rules within a given mode have the same overall purpose, each rule can be expressed more concisely; the overall purpose can be procedurally encoded into the interpreter for that particular mode.

The next thing which we notice about the protocol is that there is no obvious relationship between the specifications (figure 3) and the code (figure 2).[3] In particular, the top level WW goal has *three* parts, but its code has first *two*, and later *four* steps. A human tutor might say that this student has undergone a *reformulation process, whereby the problem description has been modified*, replacing the original two subgoals of drawing a roof and a pole by the single subgoal of drawing a tree (figure 4). The *reason* for doing this is that the student has already written a tree program, which can now be used as a subprocedure. The model formalizes the human tutor's insight using a data structure called the *plan*.

Figure 4.
Reformulating the Wishingwell in Terms of a Tree



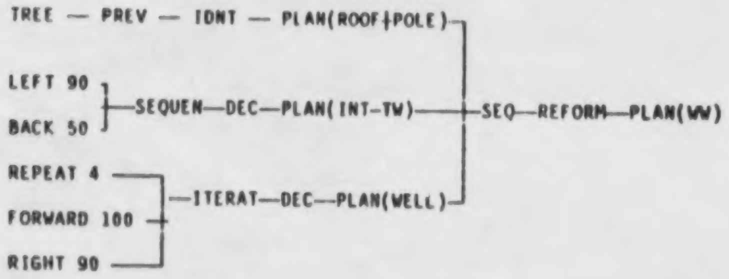
The model classifies plans according to a particular taxonomy (figure 5). At the most general level, there are three categories of planning strategies: <u>Identification</u> with previously solved problems; <u>Decomposition</u> into simpler sub-problems; and <u>Reformulation</u> of the problem description in other terms. At more detailed levels the model provides elaborations of these techniques, specifying, for example, methods for recognizing interactions among sub-goals, and deriving a linear solution. Corresponding to each plan type is a rule specifying how to refine goals of that type.

This lends precision to the notion of the plan of a program. The plan is an hierarchical representation of the design choices among alternative plan types which led to the code (figure 6). A <u>chart</u> ([Kaplan 1973], [Kay 1973]) is used to capture the notion of alternative partial plan versions. Its relevance here is that Spade-0 enables the student to access and modify this data structure, which can facilitate debugging or extending the program.[4]

Figure 5. A Taxonomy of Plans

```
                     |-- PRIMITIVE
     |-- IDENTIFY ---+
     |               |-- PREVIOUSLY DEFINED PROCEDURE
     |
     |                                |-- SEQUENTIAL
     |               |-- CONJUNCTION +
     |               |                |              |- DECOMPOSITION
     |               |                |-- NONLINEAR +
     |               |                               |- COMPOSITION
PLAN -- DECOMPOSE --+
     |               |                |-- ITERATION
     |               |-- REPETITION -+
     |                                |-- RECURSION
     |
     |                                |-- REGROUP
     |               |-- EQUIVALENCE +
     |               |                |-- GENERIC<->EXPLICIT
     |-- REFORMULATE +
                     |                |-- SPECIALIZE
                     |                |
                     |-- SIMPLIFY ---+-- GENERALIZE
                                      |
                                      |-- ANALOGY
```

Figure 6. Plan for WW



The model posits that certain types of bugs are an inevitable consequence of rational planning under limited resources and incomplete knowledge. Hence, a model of diagnostic and repair techniques complements the planning model. The diagnosis model traces the underlying causes of program malfunctions to specific erroneous choices during plan refinement mode.

The model is incomplete: a more thorough model would include additional modes, such as a phase of criticizing the program after its successful completion. Even if augmented in this way, it would be premature to claim any psychological support for the model. However, informal but detailed analysis of over a dozen student protocols has been encouraging. The resulting plan structures yield insights into the problem-solving behavior which: (a) would not otherwise be apparent; and (b) are consistent with introspective and retrospective accounts [Miller & Goldstein 1977b].

Spade-0 was implemented to explore various aspects of this model. Other papers by Ira Goldstein and the present author (e.g., [Miller & Goldstein 1977a]) provide perspective on the overall project. The model also underlies the design for an automatic programming system [Goldstein & Miller 1976b], and a protocol analyzer for elementary programming [Miller & Goldstein 1976d]. Each of these, when implemented, will be incorporated into the Spade environment to enhance its tutorial capabilities (as will be explained later).

Spade-0 is a First Step Toward a Programming Tutor

To illustrate the application of the model to assisting beginning programmers, we present excerpts from a session with Spade-0. We assume the student has chosen to draw the Wishingwell depicted earlier.

Q1. What is your name?
A1. > John Smith

Q2. Hello, John. Good to see you again.
Do you want me to get your work from the file (JSWORK 35)?
A2. > No

Q3. Ok. What name for the procedure you wish to plan today?
A3. > WW

Q4. What type of plan would you like to use?
A. Identify as previously solved problem.
B. Decompose into sub-problems.
C. Reformulate into alternative problem.
A4. > SEQUENTIAL

This interaction illustrates one of the many sources of difficulty in implementing practical computer tutors. A typical computer aided instructional system would reject the student's answer here, because it is not one of the choices which were offered. Yet, a human tutor would have no trouble understanding the student's response in this situation (despite an inherent ambiguity). Spade-0 addresses this by an abbreviation capability, which finds the shortest path between the current rule and the unexpected answer.

Q5. Ok. I am assuming PLAN -> DECOMPOSE -> SEQUENTIAL. What are the main steps in this sequential plan?
A5. > Well Pole Roof

Q6. Do you want to include a SETUP step?
A6. > Later

Ok, John. I am postponing the [setup] of WW until later.

Q7. What now?
A7. > Goto the second [interface] of the sequential plan.

Q8. Do you want to include this optional interface?
A8. > Yes

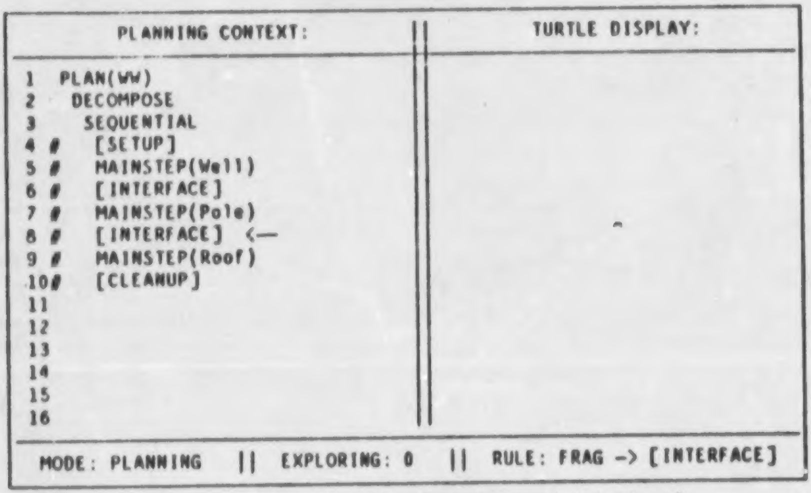
One of the difficulties encountered in early versions of the system was that even trivial graphics programs have a considerable number of nodes in the chart structures which represent their plans. Because of this, there seemed to be no convenient way for the user or the system to unambiguously refer to nodes in the evolving structure. Requiring the student to label each node by its unique internal name (e.g., "G00037") was unacceptable. Yet, a design objective was that the student should be able to walk around freely in the structure, examine the choices at various nodes, edit nodes, attach assertions to nodes, and so on. This problem was resolved by two techniques: (1) a more elaborate parser was devised for disambiguating references to nodes; and (2) an advanced display capability was implemented, allowing the plan to appear on the screen, and the current node to be brightened.

Figure 7 shows the display screen at this point in the session. Several "windows" are employed. The teletype part of the interaction occurs on the lower half of the screen. The user's turtle picture is drawn in the upper right corner (currently only the cursor is visible in this region, since no drawing commands have yet been issued). Relegating the turtle display to one corner -- rather than allowing it to usurp the entire top of the screen as in standard Logo -- leaves several areas free for pedagogical purposes. The user's skeletal plan is displayed in the upper left, and a line of "mode" information is available in the center. As the plan is refined and modified (by the addition or deletion of nodes), its representation on the screen is updated accordingly. The current goal is brightened (indicated by an arrow), and each pending goal is flagged by the symbol "#". The mode line indicates the current rule mode, the exploration level (to be explained below), and a shorthand representation for the rule governing the current interaction.

Note how the system exhorts the user to adhere to an hierarchical refinement process. The student is not distracted from understanding his problem and designing a high level plan, by a necessity to communicate with the machine in low level string-oriented editing commands. Nor do errors of programming language syntax obscure the student's task, since no code is ever typed by hand. Instead, the final program is computed from the student's answers to the multiple choice questions; hence typographical errors can be trivially noticed and immediately corrected.

At each step, Spade-0 chooses an appropriate next goal to pursue; but there is no requirement to accept its choice; interactions 6 and 7 illustrate the student ignoring the system's advice, and selecting a different goal to pursue instead. This illustrates how our desire to obtain feedback about the underlying model constrained the implementation. Although the system always encourages obeying the model, features are provided to allow violating it; and all such features are available at all times.

Figure 7. Spade-0 Display Screen During Planning of WW



```
        PLANNING CONTEXT:          ||        TURTLE DISPLAY:

 1  PLAN(WW)
 2    DECOMPOSE
 3      SEQUENTIAL
 4 #      [SETUP]
 5 #      MAINSTEP(Well)
 6 #      [INTERFACE]
 7 #      MAINSTEP(Pole)
 8 #      [INTERFACE]  <—
 9 #      MAINSTEP(Roof)
10#       [CLEANUP]
11
12
13
14
15
16

 MODE: PLANNING  ||  EXPLORING: 0  ||  RULE: FRAG -> [INTERFACE]
```

Q8. Do you want to include this optional interface?
A8. > Yes

Enforcing these constraints requires some ingenuity. At the top level, the system contains a flexible driver loop, which invokes a scheduler to suggest which goal to pursue. Associated with each goal is a specific rule which it obeys; typically the rule will generate subgoals, each with a specified rule governing <u>its</u> expansion. The driver loop incorporates a powerful escape mechanism that allows the user to ignore the current goal at any time, activating a feature instead. This requires recording the status of each interaction in a global database, so that no information crucial to that interaction is lost if volatile storage (e.g., the push down stack of function invocations) is returned. As a result of this program organization, the dialogue may be interrupted at any point to perform any arbitrary computation; when resumed, Spade-0's knowledge of the user's goals is intact.

Spade-0's flexibility has enabled us to experiment with alternative configurations of features and rules. The system has already undergone several iterations of informal testing and revision, resulting in better formulations of the planning rules, and an improved understanding of the merits of various features. The LATER feature illustrated in the above scenario provides an example.

One of the modes in the underlying model is responsible for choosing which subgoal to work on next. The choice is determined by reconciling the competing indications of a weighted set of *preference rules*. Initially the preference rules favored a primarily top-down, left-to-right solution order. In informal experiments, graduate student subjects found this too restrictive, and used the LATER feature surprisingly often. For example, subjects invariably prefer to solve the POLE of WW *before* the preceding interface step.

To correct this, additional rules were added. At first, the new rules seemed *ad hoc*: other things being equal, solve main steps before interfaces; prefer direct neighbors to non-adjacent goals; prefer left-to-right order; prefer simple goals to more complex goals. In reviewing the new rules, however, a general principle emerged: *design decisions should be made in that order which minimizes the probable scope of future modifications*.

Of course, this *Least-Scope* principle cannot be implemented directly, since one cannot tell in advance what is likely to change. The new rules could not be replaced by the principle, but they could be *understood* as heuristics which tend to approximate it. Consider one of these new rules -- main steps before interfaces -- how does this approximate Least-Scope? One of the most common bugs in sequential plans, prerequisite clobbers bother goal [Slussman 1973], involves an incorcorrect ordering of main steps. The repair technique is to re-order the main steps: the bodies of the main steps are unaffected; but interfaces must be re-designed. Likewise, "Top-Down Programming" is an excellent approximation to Least-Scope. But it is only one of several heuristics; in isolation, it is inadequate. The Least-Scope principle in turn suggests further refinements which might otherwise be overlooked; e.g., prefer simple goals to more complex goals *except* when the more complex goal constrains the simpler one.

Consider the following situation. The user is in the process of defining wishingwell. However, prior to completing either the initial SETUP or the first INTERFACE, the user requests execution of the program, obtaining an incorrect picture (figure 8). The scenario below illustrates the assistance Spade-0 can provide.

Q9. What now?
A9. > RUN WW

Q10. ⟨Running WW ... Done.⟩ What now?
A10. >Debug

Debugging WW. The top level of WW contains 7
design decisions which could be faulty.
There are warnings attached to several nodes
in the plan for WW.

According to the model, the planner leaves warnings on nodes for which the design decision was arbitrary, to guide later debugging. This is reflected in Spade-0, in that the user may attach a warning to a node as a reminder of a possible bug. The system also adds warnings on its own when various conditions are satisfied. For example, use of the LATER feature results in a system-generated warning that a node may be incomplete. The presence of a warning increases Spade-0's suspicion that a given node is the locus of an error; and user-generated warnings are given greater weight.

Q11. Do you want to try to diagnose bugs in WW?-
A11. > Yes

Q12. What now?
A12. > Warnings?

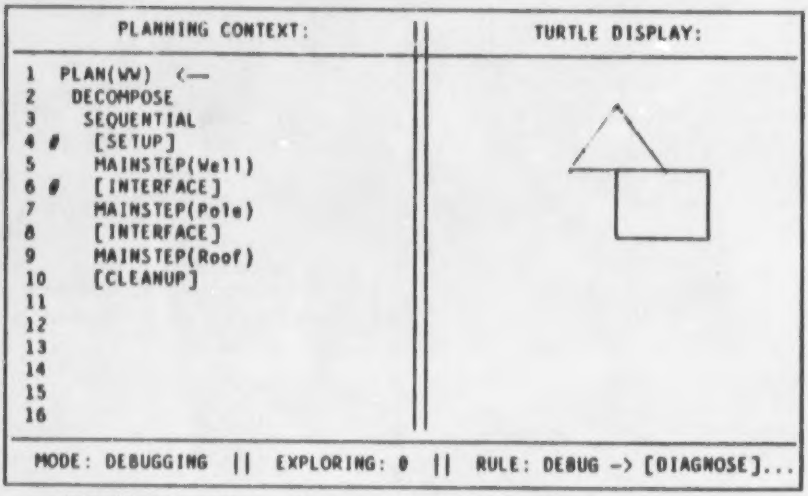
I am assuming DIAGNOSE -> PLAN? -> WARNINGS?

The WW plan contains the following warnings:
  The [setup] goal of the sequential plan has
  not been completed.
  The first [interface] has not been completed.

Q13. What now?
A13. > Repair the [interface] between pole
     and well.

I see, John. You have concluded that the [interface] between pole and well has a sematic bug: specifically, the missing step, which is syntactically optional, is needed for this particular problem. Completing...

Figure 8. Spade-0 Display Screen During Debugging of WW



```
              PLANNING CONTEXT:              ||          TURTLE DISPLAY:
1  PLAN(WW)  <—
2   DECOMPOSE
3    SEQUENTIAL
4 #   [SETUP]
5     MAINSTEP(Well)
6 #   [INTERFACE]
7     MAINSTEP(Pole)
8     [INTERFACE]
9     MAINSTEP(Roof)
10    [CLEANUP]
11
12
13
14
15
16
MODE: DEBUGGING  ||  EXPLORING: 0  ||  RULE: DEBUG -> [DIAGNOSE]...
```

```
Q10 (Running WW ... Done.) What now?
A10. > Debug
Debugging WW.  The top level of WW contains 7 design decisions
which could be faulty.  There are warnings attached to several
nodes in the plan for WW.
```

In order for a student to converse successfully with Spade-0, a certain amount of terminology must be mastered. Hence, the system contains a rudimentary tutoring module, which attempts to develop a shared vocabulary about plans, bugs, and debugging techniques. The following scenario illustrates how new terms are introduced when appropriate situations arise.

Q11. The rule is : DEBUG -> [DIAGNOSE] + [REPAIR]. Do you want to try to diagnose bugs to WW?
A11. > Yes

Q12. You have never used DIAGNOSE before. Would you like an explanation?
A12. > Yes

Diagnosis means figuring out which node in the plan represents an incorrect choice. There are

> are four sources of evidence. (1) Aspects of the plan itself, such as warnings attached to nodes, may be examined. (2) The problem description can be checked, to ensure that a code segment has been written to achieve each goal. (3) A trace of the running processes can be used to detect such mistakes as undoing previously accomplished steps. (4) The code can be scanned for unusual patterns such as consecutive calls to FORWARD with no rotations in between.

It should be emphasized that the tutoring module is extremely primitive. It consists of a small collection of ad hoc rules which: (1) decide when to interrupt the student, and (2) select one of the explanations from a data base of pre-stored messages.

Suppose that the same student, still working on parts of wishingwell, wishes to try an entirely different plan. However, in order to encourage such exploration processes, many students need reassurance that the current state of the solution can be restored, should the alternative plan prove unsuccessful. Hence, although such human-engineering features may seem to be of only minor theoretical intersest, their provision in Spade-0, as illustrated below, seems essential.

Q14. What now?
A14. >EXPLORE

Ok. The current solution has been recorded as exploration 3. To get it back, type "Restore 3".

Q15. What now?
A15. >Goto the decompose node of WW

Ok. The active plan for the decompose of WW is SEQUENTIAL.

Q16. What now?
A16. >Edit

Editing the decompose of WW.
Only one choice is left: Decompose -> Repetition.

Q17. What now?
A. Iteration

```
    B. Recursion
A17. > A

Q18. What now?
     ...
A18. > RESTORE

The last saved exploration was version 3.  I am
restoring version 3 of your plan for WW, John.
The sequential version of the decompose for WW
is now active.

Q19. You were pursuing the first interface of
     the sequential plan.
     What now?
```

The effort involved in exploring unsuccessful alternate plans is rarely wasted. The efficient representational scheme allows all versions of all plans developed during previous explorations to remain available, and these "hung" subplans often turn out to be useful later on.

The importance of such system features is that they provide a flexibility which goes beyond our current theoretical framework. By monitoring the use of such features, we can discover ways in which the underlying model needs extension or revision.

Spade-0 is an Experimental Tool

We expect advanced version of Spade to be of practical value in tutoring beginning programmers.[5] However, Version 0 is primarily intended as an experimental vehicle for exploring various facets of the underlying model of planning and debugging.

The following issues of design methodology are currently being investigated using Spade-0:

a. Do users find the planning rules adequate; or are there planning concepts which cannot be expressed in this framework? Is there a single optimal set of rules, or should the system be tuned to each individual's "style".

b. Do the plan structures generated by Spade-0 serve as useful documentation, aiding one programmer in understanding and modifying programs written by another?

c. How many rules remain the same in moving from one task domain to another? We are in the process of designing problem solving models for the "blocks world," a set theory world, and an elementary calculator world.

d. The "Later" feature (illustrated in interaction 6 of section 4) allows the user to deviate from the system's default solution order. Do novice programmers in fact make more frequent use of this feature than experts? Does Least-Scope predict the contexts in which experts find Top-Down programming inappropriate?

e. How effective is the system in giving supportive practice in structured programming? Can its effectiveness be attributed to grater articulation of planning and debugging strategies?

The process of implementing and testing Spade-0 has already resulted in several refinements to our model. It led to the discovery of Least-Scope, a principle governing the order of design decisions which is more subtle than Top-Down. It changed our view of debugging, from an early grammar-based representation, to the current formulation as a separate mode, in which transformational procedures map plans into plans. Most recently it has focused our attention on the role of exploration. We are currently attempting to adapt the model to incorporate this phenomenon, and to formalize the ways in which other flexibilities in the system's features are employed.

Limitations and Extensions

Spade-0, although useful as a tool, is not a practical tutor in its current form. Its limitations are instructive, however, providing insight as to the additional knowledge and expertise which would be required for a successful tutor. We will examine three such limitations.

1. Although a problem specification language for turtle graphics [Goldstein 1974] plays a central role in the underlying model, Spade-0 does not use information in the problem description. This severly restricts its range of planning advice and debugging assistance, a limitation which cannot be surmounted merely by better human-engineering, by extending the collection of plan types, or by adding new features

within the existing framework.

To address this difficulty, we have designed and are now implementing Spade-1, an improved environment, based on the use of Goldstein's predicate language to specify the problem description. The following proposed scenario illustrates how this extra information will enable a natural extension to the current system.

Q20. What is the name of this procedure?
A20. > WW

Q21. Please enter the problem description for WW.
A21. > WW is Roof, Pole, Well
> Roof is-a Triangle
> Pole is-a Line
> Well is-a Square
> Roof is-above Pole
> Pole is-above Well
> Roof is-connected-to Well
> Well is-connected-to Pole.

Q22. Ok. The WW problem is well-defined. What now?
A22. > Roof Pole Well

These steps are Parts in the problem specification.

Q23. Are you pursuing a sequential decomposition plan?
A23. > Yes

Q24. Ok. What now?
A24. > ADVICE

WELL is described as a square in the WW specification. The answer library contains a procedure, RECT, which draws rectangles. A square is a kind of rectangle. Hence, I suggest trying an identification involving RECT as the plan for WELL.

In the extreme case where the user requests advice about every decision, a module capable of performing "automatic programming," albeit within a limited domain, is required. For each plan type, the rules for propagating sematic and pragmatic annotation ("purpose" and "reason" assertions) downward through the plan must be specified. Goldstein & Miller [1976b] present an early version of the design for such a module; a revised description will be presented in a forthcoming doctoral dissertation by Miller.

2. Even with the extension to Spade-1, the allocation of effort to detailed articulation of the plan would quickly begin to seem excessive as the student became more experienced: Spade-0's simple abbreviation capabilities will not suffice. Hence, we have designed Spade-2, a further improvement of the environment, based on the use of a chart-based parser ([Kaplan 1973], [Kay 1973]) to infer the user's plan from fragmentary and implicit information at varying levels of detail. The following proposed scenario illustrates how Spade-2, with its plan parsing module, would represent a further extension of the system described here.

```
Q25. The current problem is Wishingwell.
     Its parts are Roof, Pole, and Well.
     What now?
A25. > Square

Q26. I am assuming: Well has been solved by
     calling Square.
     What now?
A26. > Forward 100

Q27. Ok.  What now?
A27. > Triangle

Q28. I am assuming:  Roof has been solved by
                     calling Triangle.
     I am assuming:  (Forward 100) was in-line
                     code for Pole.
     I am assuming:  the overall plan type is
                     Sequential.
     The interfaces are still pending...
```

A preliminary design for the plan parser was presented in [Miller & Goldstein 1976d]. Whether efficient implementation can surmount the inherently combinatoric nature of the plan recognition problem remains an open question; related work by Rich et al. [1978] and Waters [1976] is encouraging.

As the environment increases in sophistication, it becomes possible to contemplate reducing the highly structured style of interaction. For example, the user interface for Spade-2 could be engineered to closely resemble standard Logo. However, various advise-giving commands would be available, based on having inferred the student's plan. This presents a paradox: it is by no means obvious that a less structured environment is, in general, more conducive to learning. For example, the structured prompting of Spade-0 might be ideal for inculcating the virtues of outlining, in an essay-writing environment. Likewise, the optimal style of interaction may vary widely across individuals.

3. This dilemma draws attention to the most serious problem confronting Spade-0 and most similar systems: paucity of pedagogical theory. To approach human quality, computer tutors must become sensitive to such issues as: (i) "what constitutes a good hint?" (ii) "should the computer spontaneously interrupt the student to provide remedial advice?" (iii) "how often should the tutor intervene?" (iv) "how much should it say?" (v) "when should it say it?", and (vi) "how should it say it?". Considerable theoretical and experimental work will be necessary before such questions can be answered with any confidence.

Related Work

Spade-0 is one of several recent efforts to apply artificial intelligence (AI) theory and techniques to the design of personal learning environments. The most closely related projects include: the SOPHIE electronics tutor [Brown et al. 1974], the SCHOLAR geography tutor [Collins et al. 1975], the WEST arithmetic tutor [Burton & Brown 1976], Sleeman's[1977] tutor for interpreting nuclear magnetic resonance spectra, the WUMPUS Advisor of Carr & Goldstein [1977] the FLOW programming tutor of Gentner & Norman [1977], and Genesereth's [1977] MACSYMA Advisor.

A second area of related work concerns the construction of computer assistants for programmers. Most other projects have focussed on "apprentice" systems for expert programmers; this is primarily a difference in emphasis. A recent flurry of activity was initiated by [Teitelman 1970]. [Winograd 1973] went on to further define the problem, which has since been explored by [Hewitt & Smith 1975], and pursued by [Rich & Shrobe 1976]. Rich et al. [1978] are currently investigating assistant environments for developing Lisp programs such as hash coding schemes, and Fortran programs such as are found in the IBM Scientific

Subroutine Library.

Many of the objectives and ideas leading to Spade-0 are descended from research scattered over a much broader arena. The need for a unified account of planning and debugging became apparent in recent dissertations by Sussman [1973], Goldstein [1974], and Sacerdoti [1975]. The attempt to provide explicit normative rules of design methodology has analogues in the structured programming movement (e.g., [Dahl et al. 1972], [Dijkstra 1976]). The search for psychologically viable information processing models of the design process is closely related to work by Newell & Simon [1972], and Brooks [1975]. The goal of providing a programming curriculum which emphasizes problem solving concerns is inspired by the writings of Polya [1965] and Papert [1971].

Conclusion: The Role of Limited Didactic Systems

Even when the task is a simple arithmetic game such as West [Burton & Brown 1976], designing an "artificially intelligent" computer tutor is an arduous task, beset by shortcomings of theory and technique. In order to progress to more challenging domains such as computer programming, intermediate subgoals must be established. Didactic systems with limited objectives can lay the foundations for later work on more sophisticated and complete tutors. They can drive theory evolution, and suggest alternative applications. They can direct attention to serious issues previously unnoticed, and away from bogus issues previously thought crucial. Moreover, the pedaogical mileage obtainable from a limited system can often be surprising.

The development of a structured planning and debugging environment illustrates how such an intermediate system can help. It has provided a useful research context for studying the knowledge which a programming tutor would need to convey, resulting in several contributions to date.

a. The model underlying Spade-0, which codifies several categories of programming skill, has been elaborated and refined in several ways. The revised model unifies early theories of planning and debugging, and addresses additional questions such as the order of pursuing subgoals during planning. It appears to provide a cogent analysis of simple programming, and a coherent framework for extending the analysis to more complex programming.

b. Spade-0, an environment for elementary programming, attempts to assist novice programmers by emphasizing the plan instead of the code. It introduces the student to a vocabulary for describing plans, bugs, and debugging strategies; it provides a learning-by-doing context for exercising an articulate problem-solving style; and it organizes the student's program development process by supportive prompting and mixed-initiative menu-selection editing.

c. A set of experimental techniques have been devised for exploring models of design and learning processes in limited didactic systems. Those skills which are relevant to a given task are isolated, by systematically varying the configuration of features and rules, and by continuously monitoring the use of each capability in particular contexts.

We hope to make further contributions in the following directions: pursuing our experiments with Spade-0; extending the depth and breadth of the underlying model; and implementing Spade-1 and Spade-2. Whereas Spade-0 serves primarily as an experimental tool, Spade-1 -- which will be operational in the near future -- should approach the threshold of practical utility. Spade-2 will take a bit longer; but we expect it to surpass that threshold.

The application of artifical intelligence techniques to the design of personal learning environments is an emerging enterprise with considerable theoretical interest and exciting practical potential. Limited didactic systems provide a medium for cultivating more refined cognitive and pedagogical theories; better theories in turn sustain less restricted and more effective systems. Such systems will increasingly supplement the guidance provided by human teachers; whereupon the manner in which knowledge is disseminated, in every field from computer programming to English composition, will be radically transformed.

References

Brooks, R., May 1975. A Model of Human Cognitive Behavior in Writing Code for Computer Programs. Carnegie-Mellon U., Report AFOSR-TR-1084.

Brown, John Seely, Richard R. Burton, and Alan G. Bell, March 1974. Sophie: A Sophisticated Instructional Environment for Teaching Electronic Troubleshooting (An Example of AI in CAI). Final report, Bolt, Beranek, and Newman Report 2790 (A.I. Report 12).

Burton, Richard R., and John Seely Brown, February 1976. "A Tutoring and Student Modelling Paradigm for Gaming Environments," in R. Colman and P. Lorton Jr. (eds.), Computer Science and Education (Advance Proceedings of the Association for Computing Machinery Special Interest Groups on Computer Science Education and Computer Uses in Education Join Symposium, Anaheim, Cal.). SIGCSE Bulletin, Volume 8, Number 1, SIGCUE Topics Volume 2, pp. 236-246.

Carr, Brian, and Ira P. Goldstein, February 1977. Overlays: A Theory of Modelling for Computer Aided Instruction. Massachusetts Institute of Technology Artificial Intelligence Laboratory Memo 406 (Logo Memo 40).

Collins, Allan, Eleanor H. Warnock, Nelleke Aiello and Mark L. Miller, 1975. "Reasoning From Incomplete Knowledge", in D. Bobrow and A. Collins (eds.), Representation and Understanding: Studies in Cognitive Science. New York, Academic Press, pp. 383-415.

Dahl, O. J., E. Dijkstra, and C. A. R. Hoare, 1972. Structured Programming. Academic Press.

Dijkstra, Edsger. 1976. A Discipline of Programming. Prentice-Hall, Englewood Cliffs, N.J.

Genesereth, Michael R. 1977. "An Automated Consultant for MACSYMA," in Proceedings of the Fifth International Joint Conference on Artificial Intelligence (Volume II), p. 789.

Gentner, D. R., and D. A. Norman, 1977. The Flow Tutor: Schemes for Tutoring. Center for Human Information Processing, Technical Report 7702, San Diego.

Goldstein, I., September 1974. *Understanding Simple Picture Programs*. Massachusetts Institute of Technology, Artifical Intelligence Laboratory, TR 294.

Goldstein, I., and M. Miller, December 1976b. *Structured Planning and Debugging: A Linguistic Theory of Design*. Massachusetts Institute of Technology, Artificial Intelligence Laboratory, Memo 387.

Hewitt, C., and B. Smith, 1975. "Towards a Programming Apprentice," in *IEEE Transactions on Software Engineering*. Volume I, Number I.

Kaplan, R., 1973. "A General Syntactic Processor," in R. Rustin (ed.), *Natural Language Processing*. Courant Computer Science Symposium 8, NY, Algorithmics Press.

Kay, M., 1973. "The MIND System," in R. Rustin (ed.), *Natural Language Processing*. Courant Computer Science Syposium 8, N.Y., Algorithmics Press.

Miller, M., and I. Goldstein, December 1976d. *PAZATN: A Linguistic Approach to Automatic Analysis of Elementary Programming Protocols*. Massachusetts Institute of Technology, Artificial Intelligence Laboratory, Memo 388.

Miller, M., and I. Goldstein, August 1977a. "Structured Planning and Debugging," in *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*. Cambridge, Mass.

Miller, M., and I. Goldstein, October 1977b. "Problem Solving Grammars as Formal Tools for Intelligent CAI," in *Proceedings of ACM77*.

Newell, A., and H. Simon, 1972. *Human Problem Solving*. Englewood Cliffs, N.J., Prentice Hall.

Papert, S., 1971. *Teaching Children to be Mathematicians Versus Teaching About Mathematics*. Massachusetts Institute of Technology Artificial Intelligence Laboratory, Memo 249.

Polya, G., 1962, 1965. *Mathematical Discovery*. Volumes I and II. N.Y., John Wiley and Sons.

Rich, C., and H. Shrobe, December 1976. Initial Report on a Lisp Programmer's Apprentice. Massachusetts Institute of Technology, Artificial Intelligence Laboratory, TR 354.

Rich, C., H. Shrobe, R. Waters, G. Sussman, and C. Hewitt, January 1978. Programming Viewed as an Engineering Activity. Massachusetts Institute of Technology, Artificial Intelligence Laboratory, Memo 459.

Sacerdoti, E., September 1975. "The Nonlinear Nature of Plans," in Proceedings of the Fourth International Joint Conference on Artificial Intelligence. Tbilisi, Georgia, USSR.

Sleeman, Derek H., August 1977. "A System Which Allows Students to Explore Algorithms," in Proceedings of the Fifth International Joint Conference on Artificial Intelligence. (Volume II), pp. 780-786.

Sussman, G., 1973. A Computational Model of Skill Acquisition. Massachusetts Institute of Technology, Artificial Intelligence Laboratory, TR 297.

Teitelman, W., April 1970. "Toward a Programming Laboratory," in Buxton and Randell (eds.) Software Engineering Techniques.

Waters, R. C., August 1976. A System for Understanding Mathematical FORTRAN Programs. Massachusetts Institute of Technology, Artificial Intelligence Laboratory, Memo 368.

Winogard, T., November 1973. "Breaking the Complexity Barrier (Again)," in Proceedings of the ACM SIGIR-SIGPLAN Interface Meeting.

NOTES

1 This paper describes joint work carried out in collaboration with my doctoral thesis advisor, Ira P. Goldstein, whose significant contributions to its form and content are gratefully acknowledged. I would also like to thank John Seely Brown and Derek Sleeman for editorial suggestions significantly improving the presentation. A slightly revised version of this paper is to appear in the International Journal of Man-Machine Studies (January 1979). The research was performed at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. It was supported in part by the National Science Foundation under grant SED77-19279.

2 More precisely, we have devised a framework within which we model this and similar protocols. One model within that framework corresponds to the "ideal" student; other models are viewed as perturbations from the ideal student, and represent individual students. The current system does not use individual student models; however its design was derived using this overall framework, and a planned system will include provisions for automatically inducing individual student models.

3 Of course, there is no way to be certain that the student viewed the problem exactly as we have represented it in the specifications. However, the informal evidence in typical verbal transcripts tends to support this description.

4 Assertions must be attached to various nodes of this structure in order to fully describe the plan. This "semantic and pragmatic annotation" is specified by the model but not implemented in Spade-0. (We will return to this point in a later section on the limitations of the current system.)

5 We also believe that some of the ideas and techniques will prove helpful in the development of bookkeeping assistants and "apprentices" for professional programmers.

List of Figure Captions

1. Wishingwell: A Typical Introductory Logo Project
2. A Typical Student Solution
3. Specifications for Wishingwell
4. Reformulating the Wishingwell in Terms of a Tree
5. A Taxonomy of Plans
6. Plan for WW
7. Spade-0 Display Screen During Planning of WW
8. Spade-0 Display Screen During Debugging of WW

Cost-Effective Implementation of Music Notation on CRT Terminals

Richard L. Hamilton
Computing Center

Rosemary N. Killam
School of Music

North Texas State University

INTRODUCTION:

In 1977 North Texas State University (NTSU) made a commitment to provide a comprehensive program of computer-assisted instruction (CAI) in music theory for its undergraduate music majors. NTSU has in excess of 600 students enrolled in freshman and sophomore ear-training classes, all of whom are potential beneficiaries of the CAI project. The goal has been to offer each of these students an average of three half-hours per week of computer-assisted drill and practice in a cost-effective manner.

The NTSU School of Music attracts its student population from across the nation. The school has long been known for its predominance in jazz, a field not generally covered by systematic pre-college instruction, and has students with extremely diverse backgrounds and prior training. The CAI program is designed to meet the needs of this varied group of students, most of whom are planning professional careers in music.

BACKGROUND:

The project initiated experimental student use in September, 1977 with one prototype terminal housed in the Computer Science Department. One freshman ear-training class of 20 students received one half-hour of ear-training CAI per student per week. During the second semester of the 1977-78 school year the prototype terminal was moved to one of the School of Music's buildings and was opened on a limited basis to freshman ear-training students. Construction of 10 additional instructional units was started then and is now nearly complete.

Parts shortages and ordering complications delayed completion of the first terminal for some weeks, but modifications necessary for music notation were installed on the prototype terminal on July 5, 1978. Expansion of the system has proceeded since then, with four more terminals available to students on October 30, 1978 and another four ready by the beginning of December, 1978, all with music notation capability.

NTSU's HP2000 timeshared BASIC computer is used to present lessons, with musical examples being performed by a microprocessor-based sound production system (AMUS--Automatic MUsic System) developed by Dan W. Scott, chairman of the NTSU Computer Science Department (see Hamilton and Scott, 1978 and Bales, Hamilton and Scott, 1978). AMUS combines a Motorola M6800 microprocessor with specially designed hardware to make an economical, yet quality, instrument which is controlled by the HP2000 during the course of the lesson. The curriculum writer inserts music text, in a simple encoding language, at any point in the lesson where music is desired.[1,2]

Professor Rosemary N. Killam, NTSU School of Music, obtained an NTSU Faculty Research Grant for development of cost-effective music notation on one CRT (Cathode Ray Tube) terminal, to be completed by August, 1978. Professor James Mackey, NTSU Physics Department, and Mr. Dwight Maxim, engineer, NTSU Physics Department, volunteered a plan whereby the hardware changes necessary for music notation for one terminal could be expanded to all 10 terminals, within the budget allotted by the grant.[3]

PLANNING ASSUMPTIONS:

Planning centered on the development of a notational system, cost-effective for CAI, which would support sound curriculum strategies. The project utilizes the ADM3A terminal which is an inexpensive, durable terminal. The ADM3A uses a 5 by 7 dot character generating matrix; it has lower case capability, an addressable cursor, and a 24 line screen with an 80 character line.

Experiments with the 5 by 7 dot matrix showed that the entire symbol for such common items as eighth notes could not be generated within a single matrix. Therefore, the note stems and heads are generated by different matrices, their combination under software control. Professor Mackey generated a wide range of alternate characters, the most legible and proportional of which were chosen by the Music CAI Committee.

The notational characters replace conventional lower-case alpha characters. With the replacement of several other characters, 31 new characters were made available. This was not enough, however, to allow note stems to appear on both the upper right and lower left of note heads, as is customary in music. The committee elected to have note stems appear only on the upper right-hand side of the note heads. This allows several additional characters, such as a fermata, to be made available and permits the retention of some lower-case alpha characters frequently used in ear-training, such as m(for minor), and o(for diminished).

METHOD:

The hardware changes to the ADM3A consist of replacing the Read Only Memory (ROM) which defines the lower-case character set with a Programmable ROM (PROM) which contains the special music character set. Once the PROM has been made it is simply installed in place of the ROM.

Software was judged to be the critical aspect of the implementation. Therefore, the planning of the software was initiated in May, 1978, well before the actual hardware was available. Richard Hamilton, a programmer at the NTSU Computing Center, designed and wrote the programs.

The task of providing notation was divided into two areas:

1. The use of notation to present material to the student.
2. The use of notation by the student to enter his answers to the CAI lessons.

Presentation of notation to the student is fairly straightforward. The software uses the same encoding system developed by Professor Scott for AMUS. The curriculum writer simply prefixes the characters "$D" to any musical example in the lesson and the example is both played and displayed.

The second notational use, "graphic input" by the student, requires far more extensive strategies and thus more complex software. In contrast to this complexity, the curriculum writer uses a very simple input method, choosing the form of input as an option at the beginning of a group of questions.

For students without absolute pitch (the vast majority of music students) the actual pitch name of a note is one of the less relevant aspects of that note. It is, rather, the relationship of a given note to its predecessors and successors in a melodic or harmonic gestalt that conveys the maximum information to and for the student. Therefore, the decision was made that the student should not need to enter each pitch name. Instead, the student moves the cursor to the appropriate location on the screen, which has a staff drawn on it, types a character which represents the pitch's rhythmic value (WHQEST4), and types the return key. The program prints the note which the student has entered, in the chosen spot, and moves the cursor to the right to be ready for the next entry.

The developers had inspected previous notational systems and wished to avoid certain instructional flaws they thought present, such as:

1. If the software notifies the student that the note is right or wrong immediately, the student is encouraged to consider the musical pattern on a note-by-note basis, rather than considering the relationship of the individual note to the entire example. Students using a system which works this way tend to proceed in an almost random fashion, trying different possibilities until they stumble on the right one.

2. If, on the other hand, the software requires entry of the entire musical pattern and judges the student right only if the whole pattern is right, the student is often baffled as to the location of his error(s) and has no clear concept of how to correct them.

The NTSU system attempts to avoid both of these extremes. The student proceeds from note to note, reinforcing his concept of the pitch contour by his entry procedures. At any point, he may ask to hear the entire pattern again by typing "?". There are no limitations to the number of times he may rehear the music. He may ask for his answer to be checked at any point during the entry process by typing two slashes, "//".

The program redraws the staff, retaining only the student's correct entries. The student may then fill in the incorrect notes, bypassing correct notes with a "don't care" character. He may then ask for his answer to be checked again. The software controls how many evaluations the student may take, the exact number being determined by the curriculum author depending upon the difficulty of the material. When that number is exceeded by the student the correct answer is displayed and the student is allowed to go to a new example.

Students vary unexpectedly in their comprehension of a given example. If a student encounters unusual difficulty with one example, he is best moved on to another after a reasonable number of tries (usually no more than four). User frustration mounts when the program insists on a perfect reply before it will go to the next example. The software allows the curriculum writer to estimate this frustration point and move the student on before he reaches it (a more detailed discussion of curriculum strategy is to be presented by Rosemary Killam at the Texas Music Educators Association Conference, February, 1979).[4]

Software development followed the outline of the entire project. The first major objective was to produce a program capable of displaying a score written in the AMUS input language. Although the AMUS language is not oriented towards use with a score drawing program, it was possible to write a translating program which handles a large subset of the language. At present the only omissions are performance oriented aspects of the score, metronome markings, accents, and so forth.

The drawing program uses a similar algorithm to the one used by the program which translates scores for performance. The biggest problem was that the score language does not attach a duration to every note in a multi-voice score. The duration of any single note (or rest) is determined by the total of all durations, in any voice, between the initiation of that note and the initiation of the next note (or rest) in the same voice.

In any score, except those with only one voice, a duration character applies to all voices, resulting in a set of duration symbols which represents the times between events in the piece. Thus, a long note will almost always appear as the sum of several smaller durations. The program must keep track of each voice as it translates the score. Only after a note terminates can the program calculate its duration and write the proper symbol(s) in the correct location(s) on the staff.

A problem with implementing notation was deciding how to split up durations which can not be represented by a single note. For example, should a duration of a quarter plus an eighth be notated as a dotted quarter or as a quarter tied to an eighth? In most cases the former is preferable, but in some situations the latter may be better. The current version of the score drawing program uses the dotted quarter in all situations, except where the duration crosses a barline. This has resulted in readable and consistent scores.

RESULTS:

The NTSU music CAI project has concentrated on six major curriculum areas: rhythmic dictation, chord identification, melodic dictation, harmonic dictation, interval identification and modulations. The curriculum parallels the Ottman theory text.5

To date the notational system has been most widely used for rhythmic and melodic dictation. A large number of harmonic dictation lessons, developed before the notational system became available, will be adapted for notation in the near future.

Rhythmic dictation using the notation begins with a short review of student procedures. The student is given a sample question and the answer is shown to him on the screen. Then a brief set of reminders about the procedure is presented. The student enters his answers on a one line staff, typing the duration he hears followed by a carriage return. The program then writes the note over his entry and positions the cursor in the correct spot for the next note.

Melodic dictation lessons present a more detailed review of student entry, including moving the cursor up and down. A sample question is displayed on a 5-line staff, complete with clef, key and meter signatures, The student enters his answer on an identical, blank, staff. An example of this form of entry is illustrated in Figure 1.

Harmonic dictation, still under development, has been the most difficult to implement. The student taking dictation through the system first identifies each chord with Roman numeral notation, then enters soprano and bass lines on a grand staff. Ideally, the student should be able to enter these three items in whatever way he wishes--all together, line by line, or in any vertical or horizontal order. In practice, the student must enter the chords first and then may enter soprano and bass as desired. Figure 2 illustrates a completed answer using the notational system.

Training 600 students to use the notational system has proved to be no small task. The authors of this paper are continually amazed at the variety of ways in which the students interact with the system. Such originality on the part of the students requires continual modification and expansion to meet their needs.

The Music CAI lab is currently located in the Computer Science Department's microprocessor lab, across campus from the School of Music, until a permanent location is completed in a building currently under construction. Music monitors are available for only 35 of the 83.5 hours per week that the lab is open. During the other hours monitors from the Computer Science department run the lab.

The use of the notational system, then, must be self-explanatory through the lesson structure. The measure of success so far has been that the lessons that use notation are popular with the students and widely used.

SUMMARY AND CONCLUSIONS:

A cost-effective method of implementing music notation on CRT terminals for ear-training CAI has been successfully developed at NTSU. Although funding of this project was only for a feasibility study, the results have proven successful enough to be included in the regular ear-training curriculum.

Hardware modification by NTSU Physics department personnel provided the key to immediate implementation by lowering the cost per terminal to roughly 10% of the first estimate from an outside vendor. Software development has been extremely involved, reflecting the instructional strategies of the NTSU Theory division.

Curriculum usage of the notational system has required considerable on-line explanation of student entry methods making the system, so far as possible, self-explanatory. Lessons using the notational system have been available for student use since mid-September, 1978, and have proved to be popular and successful.

206

REFERENCES

1. Hamilton, Richard L. and Dan W. Scott. "A New Approach to Computer Assisted Instruction in Music Theory." Proceedings, Ninth Conference on Computers in the Undergraduate Curricula, University of Denver, 1978. pp 142-150.

2. Bales, W. Kenton, Richard L. Hamilton and Dan W. Scott. "Computer-Aided Composition and Performance with AMUS." Presented at the International Computer Music Conference, Northwestern University, November, 1978.

3. This project was made possible by a faculty research grant from North Texas State University.

4. Killam, Rosemary N., W. Kenton Bales, Richard L. Hamilton and Dan W. Scott. "AMUS: The Computer in Music Instruction." to be presented at the Texas Music Educators Association Conference, Fort Worth, February, 1979.

5. Ottman, Robert W. Elementary Harmony: Theory and Practice, Prentice-Hall, Englewood Cliffs, New Jersey, 1972.

Learner Interaction in Computer-assisted Instruction

Esther R. Steinberg
Computer-based Education Research Laboratory
University of Illinois, Urbana

ABSTRACT - Research that focuses on learner interaction with CAI is essential if CAI is to make an effective contribution to education. We must understand both the process of learning and the role of human-machine interaction in this process in order to achieve success.

Computer-assisted instruction (CAI) is here not only to stay, but also to increase. Cost will not prevent its expansion; educational needs will foster its growth. Although it may be considered too expensive for widespread use at present, we can already point to settings, such as the training of airplane pilots (Trollip, 1977), where costly as CAI is, it is inexpensive relative to current instructional methods. With the introduction of microprocessors, costs are expected to tumble before too many years and thus make CAI generally affordable. In addition, the educational community is deeply concerned about the steady decline in test scores on college entrance examinations and decreasing competency in the basic skills. Solutions to the problem are being seriously pursued by the highest government officials (National Academy of Education, 1978). It is inevitable that people will look to technology in general, and computers in particular to help cure some of these eduational ills.

We know quite a bit about CAI. We know that it can be motivating, that people of all ages, from 5-year olds to professional students, can learn via this medium, and that it is acceptable to both students and instructors.

However, there is a lot that we don't know. First, there are misconceptions about instructional design, such as thinking that existing pedagogical techniques can be translated directly to CAI (Call-Himwich & Steinberg, 1977). Consider the case of an instructor who employs the discovery method in the classroom with great success and decides to use this technique in CAI. He provides almost no feedback that is directly helpful, expecting the student to "discover" the answer. Much to the instructor's surprise, many students who attempt the CAI lesson do not make the discovery and in fact become quite frustrated. What went wrong? The instructor failed to take into account an essential difference between the classroom and CAI. In the classroom, the instructor did not supply the answers, but various students did so. Students who did not discover by themselves learned from responses of other students. However, students working individually at terminals did not have the benfit of classroom interaction. Therefore, in CAI, the instructor, that is the computer, must supply that help.

Second, sometimes intuition and teaching experience tell us that some aspect of CAI ought to be particularly effective for the students. One such notion is that playing games will automatically lead to the development of good problem solving strategies. Another is that allowing the student to control some aspects of his learning will improve performance. The problem is that either research does not support such assumptions (cf. Steinberg, 1977), or there is no evidence to verify them.

Finally, there is inadequate information about which of the traditional insructional variables enhance learning in CAI, and under what conditions they do so. In fact, we do not know what all the variables are. If CAI is to make an effective contribution to education, we need to identify primary variables of CAI, screen out those that affect instruction and describe what that effect is. I am not worried about a "...premature extinction..." of CAI as Kearsley (1977) has put it, but I am worried that we might throw away this opportunity to make a significant contribution to education.

We should not repeat the mistakes made by developers and researchers in instructional television. Rather, we should learn from them. In 1968, when the planners of Sesame Street were looking for information that would serve as guidelines to help them develop their program, they discovered that research in instructional television did not tell them what they needed to know. Most of the studies prior to that time produced either inconsistent results or results that were statistically insignificant (Barbatsis, 1978). The majority of the studies compared instructional television with other media, or one method of using instructional television with another. This research provided almost no information about the range of variables that affect instuctional television nor did it identify the critical variables.

I suggest that one area of research that will help provide guidelines for CAI developers is the study of the learner, his interaction with the machine, and his learning processes. We need to describe what's happening when the learner interacts with the computer. We need to understand both the process of learning and the role of human-machine interaction in this process. The discussion here will be most meaningful for, and therefore limited to, CAI lessons that are tutorial, problem solving, simulations and the like. It excludes CAI drill and practice, since other issues are more relevant to those types of lessons.

First, we need to define some of the characteristics of CAI that are different from traditional instruction and study how they can affect the learner. An important condition for successful learning, in any medium, is that the student understand the task. The student brings to a task his experience and expectations from traditional instructional modes. If the task when presented in CAI is somehow different from the traditional presentation, we cannot assume that the student will perceive this difference, nor understand the lesson in the way intended by the instructional designer. We cannot expect the student to benefit from lessons that use CAI's unique capabilities if there is a discrepancy between what the student does and what the lesson writer expects him to do.

The importance of this point was brought home to me by my research in children's problem solving strategies. I presented the following problem for children to solve. Two views of a common object were shown simultaneously. They differed in one half and not in the other (Figure 1). A third picture, which was covered, was purported to be match one of them. The

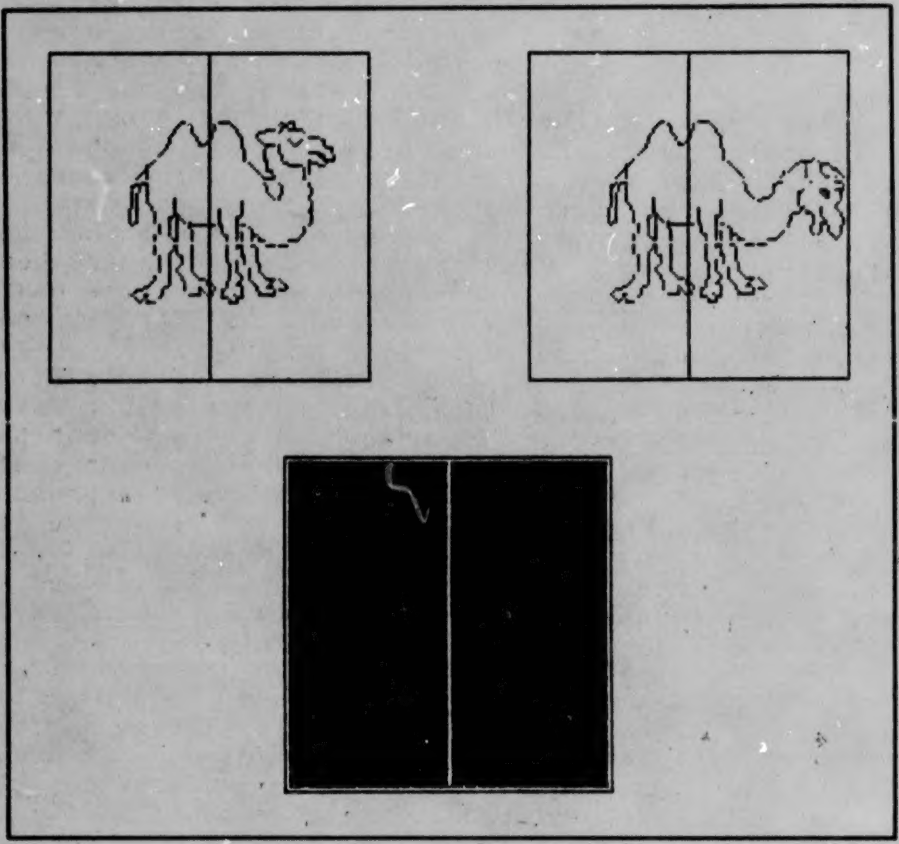


Figure 1.

child was supposed to uncover one and only one part of the hidden picture, and on that basis decide which of the views it would match. The strategy, of course, was to uncover the discriminating part. So, in the case of the camels, the correct way to do the problem was to uncover the part that showed the head. That would provide enough information to know which view would be matched. On the first day, the child saw 12 different pairs of pictures. On the second day, the child saw the same stimuli, but in a different random order. In traditional instruction, the materials stay the same from day to day. If the covered picture matched the camel with its head down the first day, chances are that the camel with its head down would be the correct match on the second day. However, this was CAI, and the computer generated the graphic displays anew at each session. The view that would be matched was randomly chosen each day. Therefore, on the second day, the correct answer might be different than it was on the first day. When the children began the task on the second day, some of them recalled which views of the pictures had been matched on the first day. To quote one 5-year old, "Oh, I remember. It was that one yesterday," as he pointed to the camel with its head up. He expected to get the right answer on the basis of memory as he could have done in traditional instruction, and not by generating a suitable strategy. He paid no attention to the usefulness of uncovering part of the covered picture, and uncovered the tail. Imagine his consternation when his guess about the match was wrong. At least 6 children spontaneously verbalized this "remember from yesterday" approach. Others probably used it too, but did not verbalize it. Of course, I explained the situation to each child. In fact, after I realized what was going on, I included an explicit explanation of the CAI manipulation in order to prevent false expectations.

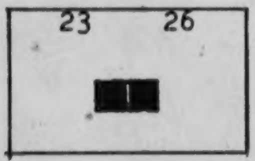
Lack of experience with CAI may lead students to completely misunderstand the nature of computer assistance. Some of the research of Rouse (1978a, 1978b) illustrates this point. He wanted to find out if computer assistance could help students develop problem solving techniques in trouble shooting tasks, and if those techniques would hold when computer assistance was withdrawn. He was interested in the kind of problem that occurs when a system, like an engine, fails and a person has to find the defective component from among the many components in each subsytem. The trouble shooter has to figure out which components to test in order to get the maximum amount of information from each test and consequently to minimize the number of parts he has to test. Rouse generated an abstract representation of this problem by presenting a graphical array of a network of nodes connected by arcs. Each time the student tested an arc in the network of nodes, the computer not only told him whether the arc succeeded or failed, it also crossed off all nodes in the network which could be eliminated from consideration as a consequence of this test. The hope was that the students would learn this strategy by observing the computer's bookkeeping technique.

Two of the groups of students who performed the task were trainees in an FAA certificate program, one just completing the 2-year program, and the other group just beginning. This was the first experience with this type of task for both groups. Performance of the second year students improved with computer assistance and was maintained when assistance was withdrawn. The results were quite different with the beginning students; they thought that the computer assistance was provided to make their problem solving task easier and that it would always be available. They did not learn the strategies because they did not understand that the computer's assistance was meant to teach them how to trouble shoot and not to do their thinking for them.

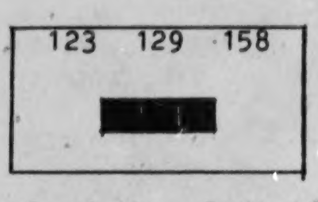
We are really just beginning to develop CAI curriculum. Some lessons are traditional both in content and pedagogical approach; others use some of the special features of CAI to present such formats as simulations and games. In either case we should not limit instructional design to traditional approaches. We need to use the findings from current research in cognitive and developmental psychology to make the traditional lessons more effective and to increase the probability that the unique capabilities of CAI will be successfully applied. One area of research that holds promise is the study of the student's learning processes and strategies. This is crucial for instructional design in general, but even more so for CAI. I do not refer to the kinds of learner variables that are associated with traditional approaches to instructional design. There, the learner is described in terms of skills and behaviors he has, or must have to do the task, as analyzed from a logical or expert point of view. So, for example, a lesson about balancing scales may require that the student be able to read at the sixth grade level and know how to multiply. These kinds of learner skills and behaviors are obviously important and should be taken into account. However, this kind of description is frequently not enough to help the student learn the material. It is also necessary to describe the way the learner learns, the strategies he uses when he interacts with the task. What is needed is an analysis "...of the structure of the learning task from the learner's point of view." (Case, 1975).

It is not necessarily the case that the learner understands the problem in the same way as the instructor. In fact, research in problem solving shows repeatedly that the novice problem solver does not do problems the same way as an expert. This has been demonstrated for such diverse populations as chess players (Chase & Simon, 1973), beginning physics students (Reif, 1977), and children learning arithmetic facts (Resnick, 1976). Before designing extensive instruction about how to solve a problem, it is important to describe how the learner tries to do the problem spontaneously (Case, 1978).

Again, I would like to refer to my own research in CAI to show how a description of the learner's spontaneous attempts to solve a problem can lead to insights about the students' understanding of the problem and how this is different from the instructor's. Consider the picture problem that was described above, with 2-digit numbers replacing pictures.



Some of the first graders solved the problem easily and also solved another another set of problems in which the stimuli were 3, 3-digit numbers. However, other



first graders who solved the 2-digit problems just as easily had great difficulty with the 3-digit. One might conclude that it was due to the fact that the 3-digit problem required more scanning and more record keeping than the 2-digit and consequently it was more difficult. In fact, this was not the explanation. The children who did both sets of problems right used the mature strategy: identify the digit that is different in all of the numbers and uncover the same relative position of the covered number. The children who had difficulty with the 3-digit problem understood that the goal was to find a discriminative digit. However, they used a poor strategy to achieve that goal. Consider the above example. The child understood that if the digit she uncovered was a 5, a 3, a 9, or an 8, she would know which number the covered one would match. So she uncovered the middle section to see if it would be a 5, not taking into consideration that it might be a 2 and she would not know which was the covered number. The lesson was programmed so that this strategy did not lead to the correct answer by chance. When the child uncovered the part that might be unique, but might not, it was not. In other words, the wrong strategy led to inadequate information. For example, in the above problem, if the child uncovered the middle part, she never saw the 5, only the 2. Then she could not tell which number would be matched. Furthermore, whichever number she guessed would be matched, the other one turned out to be the right one. So if she guessed 123, when the number was uncovered it turned out to be 129 and vice versa. Eventually, those children who had breezed through the 2-digit problems also figured out the correct strategy on the 3-digit.

To summarize, I have suggested two critical considerations for CAI instruction. First, we must describe how the learner learns; we need to understand his perspectives and strategies. Second, we must understand how the learner perceives the task as presented via CAI and how this affects his interaction with the lesson.

Reference Note

Rouse, W.B. 1978b, Personal communication.

References

Barbatsis, G.S. The nature of inquiry and analysis of theoretical progress in instructional television from 1950-1970. Review Educational Research, 1978, 48, 399-414.

Call-Himwich, E. & Steinberg, E.R. Myth and reality: Essential decisions in computer-based instructional design. Urbana, IL.: University of Illinois, Computer-based Education Research Laboratory, 1977.

Case, R. Gearing the demands of instruction to the developmental capacities of the learner. *Review of Educational Research*, 1975, *45*, 59-87.

Case, R. A developmentally based theory and technology of instruction. *Review of Educational Research*, 1978, *48*, 439-463.

Chase, W.G. & Simon, H.A. Perception in chess. *Cognitive Psychology*, 1973, *4*, 55-81.

Kearsley, G.P. Some conceptual issues in computer-assisted instruction. *Journal of Computer-Based Instruction*, 1977, *4*, 8-16.

National Academy of Education. Improving educational Achievement. Washington, D. C., 1978.

Reif, F. Problem solving in physics or engineering: Human information processing and some teaching suggestions. Submitted to *American Society for Engineering Education*, 1977.

Resnick, L.B. Task analysis for instructional design: Some cases from mathematics. In D. Klahr (Ed.), *Cognition and instruction*. Hillsdale, N. J.: Lawrence Erlbaum Associates, 1976.

Rouse, W.B. Human problem solving performance in a fault diagnosis task. *IEEE Transactions on Systems, Man, and Cybernetics*, 1978b, *SMC-8*, 258-271.

Steinberg, E.R. Review of student control in computer-assisted instruction. *Journal of Computer-based instruction*, 1977, *3*, 84-90.

Trollip, S.R. An evaluation of a complex computer-based flight procedures trainer. (Tech. Report ARL-77/AFOSR-77-1). Urbana, IL.: University of Illinois, Institute of Aviation, 1977.

Computer Based Education as a Learning Tool for Foreign Language Students

Heimy F. Taylor

The Ohio State University

About three years ago an article in the Foreign Language Annals stated the following: "CAI has yet to fulfill the promise it seemed to hold for language instruction. Its failure to do so is to a large extent the result of the failure of existing CALI (Computer Assisted Language Instruction) programs to take full advantage of the interactive capabilities of the computer."[1] The situation has not changed much since this article appeared. Although several institutions have developed CAI courses in foreign languages, the expected success has not been forthcoming, in fact some universities have discontinued their CAI programs.[2]

In spite of the fact that computers touch directly or indirectly all aspects of modern life, the cloud of apprehension or even suspicion is not yet removed when it comes to the application of this tool of technology in foreign language instruction. While the list of disciplines using computers is endless, foreign languages have been slow, almost resistant, in making use of the numerous advantages which computers have to offer. It is interesting to note that even among the younger generation of language teachers and students of today one can often encounter the feeling that the computer is a bit out of place in the humanities, especially in the teaching and learning of foreign languages. The myth still prevails that one must understand the integral mechanics of computers in order to use them. The reluctance of the profession to accept CAI as an effective tool in language instruction can be explained partially by the fact that most language teachers have had little or no training in science and technology. As humanists they often feel threatened in a world which seems to be taken over by technology. The current economic problem in the profession caused by a steady decrease in foreign language enrollment has only added to the apprehension which teachers feel towards CAI. While it cannot be denied that administrators often hope CAI will help to reduce staff, or perhaps free staff for other duties, those in the profession who have developed CAI programs in foreign language instructions are not at all interested in replacing teachers; our goal is to free them from the often boring and tedious task of conducting practice drills in grammar in order to give them time to work on other aspects of foreign language teaching, particularly conversation and pronunciation. As Nelson has stated: "We view the technology more in terms of increasing the effectiveness of ones teaching efforts, rather than as a means of dispensing with them."[3]

Perhaps the greatest reason why CAI programs have not been as successful as anticipated can be found in the programs themselves. Since a number of them do not offer anything more than what a good programmed text could do, critics have been quick in questioning their usefulness. Furthermore, the question whether CAI is indeed effective in language teaching has not been answered satisfactorily. Performance studies on a large scale are needed to determine if the use of computer programs will improve students' test scores.

Foreign langauge teaching usually has as its objective the teaching of four skills: listening, speaking, reading, and writing. It is generally agreed that listening and speaking should be stressed in the classroom and language laboratory while reading and writing are skills which could be self-taught but generally are not. Even the most experienced teacher becomes frustrated because much of the available classroom time has to be spent in drilling grammar exercises, a task which takes away valuable time from other activities. It seems that computerized instructions could be the solution to this dilemna. CAI work could relieve teachers of certain rote efforts such as going over grammar homework assignment or practicing grammar in general.

The Department of German at the Ohio State University is presently using two computer assisted instruction programs which make use of the tutorial methods. TUCO (Tutorial Computer) serves as an adjunct to the basic language courses and is offered to all beginning students on a *voluntary* basis and without charge.[4] DECU (Deutscher Computer Unterricht) is an integral and *compulsory* part of the Individualized Instruction program offered to beginning students. Both programs are designed to provide students with individualized tutorial instruction in the area of German grammar. This is accomplished with the help of the IBM Coursewriter III language and the University Systems Computer Center's IBM 370/158 computer. Students registered for CAI work can do their assignments on any of the 73 terminals located in 24 different locations on campus. They have access to terminals from 7:30 a.m. until midnight.

DECU and TUCO are not intended as a substitute for classroom or laboratory work, nor as a replacement for the teacher, but rather as a supplementary educational device which makes effective use of the principles and practices of the tutorial method, i.e. a close working relationship between teacher and student. The computer responds to the student's work with the kind of advice and hints normally given by an experienced teacher. While a teacher cannot perform in top form every day, the computer will if "fed" properly. The programs are designed so that they anticipate students' difficulties and responses; each feedback is geared to their individual answers. The memory capability of the computer offers tremendous opportunities for such tutorial teaching because it is able to store large quantities of information relating to difficulties and mistakes most frequently made by a student learning German. After signing on the student selects the segment he wants from an index, and after typing the proper number and letter he is branched to that particular section. Each segment starts with a brief text which reviews the grammar to be covered (for example, perfect tense with strong verbs). After this text is typed, the student is given brief instructions as to the several types of exercises available: fill-in-the-blanks, synthetic drills, sentence completion, sentence re-arranging, and in some very limited cases, translation. The student types in an answer and receives a response from the computer. If the answer is correct, he is branched to the next item. If his response is wrong, he is given a clue and must try to answer again until he is able to produce the correct answer. Depending on the complexity of a given section, between five and twenty-five anticipated wrong answers are programmed for each exercise or problem, and hints or responses are provided for these wrong answers.[5]

The program distinguishes between good and poor performance. If the student did well, he can go to a new segment, continue with the one on which he is working or sign off. Once a student completes a segment, he is again given a score and asked what he would like to do next. Those students who made too many errors are advised to repeat a section. This approach serves to reinforce both grammatical patterns and correct sentence structures.

There is one "crutch" which the student can use if he has responded several times and still does not understand his mistakes, or if he has no idea what the answer should be. If he types in the word "help," the computer will type out the correct answer. The student must show he has taken note of the correct answer by repeating it. Obviously, students are encouraged to use the "help" command sparingly since excessive use would defeat the purpose of the program. It seems, however, that this device has done much to alleviate excessive frustration.[6]

Computers, including corresponding programming and administration, are time consuming and costly. The usefulness of our CAI programs is challenged, therefore, from time to time by students, teachers and administrators. "Are the money and time spent on these programs really worth it?" is a question repeatedly asked by colleagues. The users of the programs are in the best position to respond to this question. In order to document students' attitude toward our CAI programs, a questionnaire was devised. During the academic year 1974/75 and in the Spring Quarter of 1976 this questionnaire was given to most of the 495 students (56 percent of our enrollment for 101 and 102), who had registered for CAI. Results reflected a direct correlation between time and effort spent with the computer and level of satisfaction. Those students who had worked regularly with CAI and had finished more than half of the program indicated the greatest enthusiasm. Written comments like: "I just think that it's a very useful study method and appreciated its use," "this course was very helpful, it was like a second teacher or tutor and gave me a chance to use what I learned," or "I felt it was an excellent program and not only found it instructional but also fun to use" were typical of regular computer users. Overall reaction to the programs was positive. Students generally felt satisfied with what they had learned. It is significant that 75 percent of our students indicated that they considered the computer to be a tutor. A comparison of the effectiveness of exercises completed via computer with those done in the classroom or at home showed that 50 percent felt the computer was better; about one third mentioned they were about the same; and the remaining 17 percent rated them as worse. The majority, over 80 percent, indicated the grammar explanations at the beginning of each section reinforced what they had learned in class and a few found them to be better. Of greatest importance to us was the question of how students felt regarding the responses from the computer when they made a mistake. Did the programmed responses help them recognize what was wrong, reinforce grammar rules, and finally guide them to the correct answers? These were our goals, and it was interesting to note that each one of these three questions received one third of the students' responses. (They could check only one answer) Only four percent said they were useless.

Considering the high level of satisfaction which students indicated, we were quite puzzled to learn that sixty percent of the students were nevertheless at times frustrated when they made a mistake. Prose comments by a few students shed some light on this problem. It was remarked that they felt frustrated when they had to look up too many words in their vocabulary list. Since the vocabulary used in TUCO does not always correspond exactly to that of the textbook used, students do indeed encounter a number of unknown words, especially in the very beginning. We also learned that students do not like the concept of having to finish a particular segment with a specific number of correct answers before being allowed to move to another section. They would have preferred to be able to shift sections at will. We have indeed changed TUCO since the results of the questionnaire; now students can go to a different section after the completion of the initial segment, regardless of their score. We are very much concerned that students master a given grammatical unit before they go on to another one and would prefer that they not turn to something new until they have received a satisfactory score. However, the frustration caused by this method outweighed the pedagogical advantages, and we therefore modified our program.

Although TUCO is designed for beginning students, each quarter students in intermediate and advanced courses ask to participate. We were quite surprised to note that every one of these advanced students had only praise to offer. They apparently depended on the computer to give them practice in basic grammar points which they no longer received in class because they were expected to know the material. Advanced students indicated that without the computer programs they would have had considerable difficulty in their respective classes; and, as some of them expressed, they might not have been able to master the course requirements. These responses seem to indicate that a greater effort should be extended to make CAI known and available to students beyond the first year level. It also pointed out that perhaps one of the greatest assets and potentials of CAI lies in remedial work. Students like the privacy, patience, consistency and instant help which they receive from the computer.

When CAI becomes an adjunct to regular classroom instruction, an important choice has to be made: Should it be a required part of the course or simply an optional learning device? The answer to this question depends in part on availability of human resources, hardware, and money. One must also decide what the overall educational philosophy of the department and the instructors is. Do you or don't you believe in prescribing your students certain activities in order to achieve specific proficiency levels? Since students were more satisfied with the program if they used CAI regularly, the possibility exists that the greatest benefit could be reaped if students were indeed compelled to use this teaching/learning aid. We feel additional research and pilot studies are needed before definite decisions can be made in our department. However, we have discovered that students who participate in our Individualized Instruction program where CAI is a compulsory part have consistently scored above ninety percent in their unit tests. Since this program is only in the beginning stage, we cannot come to any definite conclusions. We are watching students' reaction to CAI and their performance very closely, especially since at Ohio State University the German Department is the only language department which has included a CAI section in its Individualized Instruction program.

No other feature in foreign language instruction has received as much attention in recent years as the concept of individualized instruction. Many foreign language teachers view this program as the most effective weapon in combating the continuous decrease in foreign language enrollment. It can be assumed that this teaching/learning method will become a standard alternative for foreign language learning in American schools and colleges. Numerous teachers feel that with its stress on self learning, self reliance, self pacing, and self responsibility, with its offering of numerous innovative foreign language teaching techniques and approaches individualized instruction deserves to become a bona fide alternative to conventional classroom instruction.

When one thinks of individualized instruction and computerized instruction, their real and intrinsic relationship comes to mind. CAI embraces several principles which also characterize individualized instruction. Some of the major principles involved are: variable pacing, one-to-one tutoring, branching techniques, or insistence on a performance level by requiring a fixed standard of competency (80 percent at Ohio State University). Because of this "natural" partnership between CAI and individualized instruction it is very likely that computerized instruction in the future will become an important, if not an indispensable partner of individualized instruction. This is especially the case with students who need remedial work. Since individualized instruction permits students to take "their" time and CAI provides the opportunity to practice and review over and over again those areas which cause them difficulties, students will eventually be able to master the requirements of a language course. In addition, we have found that our CAI programs teach not only German grammar, they also require from students the kind of attention, concentration and self-discipline which is so important in developing successful study habits.

At the present most CAI programs only involve two of the four skills usually associated with foreign language study, namely reading and writing. However, there are already CAI programs in foreign language instruction which are geared to the four skill approach, for example, PLATO IV at the University of Illinois.

Gerhard Clausing and Rudolf Hirschmann have begun to develop a microcomputer program for German. They are including a speech synthesizer which is supposed to be capable of simulating human sounds. Clausing and his associates hope that this new type of computer will enable them to develop a wide range of materials for CAI purposes. Their pedagogical goal is to provide students with the near-realistic language experience.

At a time when learning a foreign language could be more popular than it seems to be, a teaching device like CAI should receive greater attention and scrutiny by our profession. The literature regarding computer assisted instruction in journals devoted to the improvement of foreign language teaching certainly is not overwhelming, in fact Ferdinand Ruplin, who developed a German CAI program at Stonybrook, N.Y., once mentioned that CAI is blessed by the non-proliferation of published material. Nevertheless, some studies about the effectiveness of CAI do exist or are in progress.

As far as our own DECU and TUCO programs are concerned we realize that they are not miracle workers. As must be the case with any sophisticated new tool of educational technology, there are still many improvements to be made, and a great deal of learning to be done as we gain experience. Our students certainly are satisfied with the programs as valuable learning devices. Although we have conducted some pilot studies to test the effectiveness of our programs, we need to develop a performance study in order to determine whether these programs indeed help our students to improve their test scores. At this point we are confident that both our educational goals and the needs of our students can be served more effectively by the great potential that computer assisted instruction offers.

Notes

1. G. E. Nelson, Jean Renard Ward, Samuel H. Desch, and Roy Kaplow, "Two New Strategies for Computer-Assisted Language Instruction (CALI)," Foreign Language Annals, 9 (1976), p. 28.

2. Some of the institutions which have experimented with CAI programs in foreign languages are: Stony Brook, Stanford, Dartmouth, the University of Minnesota, M.I.T., the University of Illinois, the University of Southern California, and others. A. L. Hammond reported that the successful CALI program in Russian at Stanford University was discontinued because the CALI course cost three times as much as one taught by an instructor. See: A. L. Hammond, "CAI: Many Efforts, Mixed Results," Science, 176 (June 2, 1972), p. 1006.

3. Nelson, p. 29.

4. Professor Werner Haas was instrumental in writing the TUCO program. He also gave freely of his time and assistance in writing DECU which is very similar to TUCO. It is through his dedication that the German Department at Ohio State University was able to develope a computer assisted instruction program. Finally, many of the ideas expressed in this article originated with him.

5. Our programs follow the so-called Uttal method. The sequence is as follows: a. present problem, b. request response, c. evaluate response, d. alter the flow of instructional information depending on the correctness or mistake of the response. If response is incorrect, then a. restate the problem, b. test the error hypotheses, c. generate an error detecting probe, d. classify remedial needs and provide tutorial assistance.

6. Detailed results of the attitude questionnaire will be published in a forthcoming article: "DECU/TUCO: A Tutorial Approach to Elementary German Instruction," Foreign Language Annals.

Carole A. Bagley,
Thomas Hansen
Daniel L. Klassen
Minnesota Educational
Computing Consortium

IS TCI* THE ANSWER TO EDUCATIONAL PROBLEMS?

The effectiveness of the computer as a teaching device remains to be proven.

The literature abounds with media studies[1,2] and CAI evaluations[3] which have at best shown the computer to be effective as a supplement to regular instruction or as total instruction with a specialized audience. It has not proven to be any more effective than any other medium.

Some reports show a savings in time by using CAI.[1,4] Other reports show the computer to be more effective in the tutorial and drill modes than with simulation or problem solving.[5]

If it doesn't make any difference which medium we use since they all result in the same amount of achievement, perhaps we should be selecting the lease expensive. This certainly rules out the computer for use as total instruction or it moves all TCI to cheap microcomputers.

Or, perhaps, the rationale for using the computer need not be achievement gains for all students. Are other factors more or equally important? Such factors as individualization, self-pacing, immediate feedback, teacher aide - frees teacher to do other things, computer presents material to student without forgetting a concept or without becoming tired due to repetition.

Or, perhaps we can increase achievement by using the computer differently. Drill and tutorial have been shown to increase the achievement levels of low and superior ability students.[3] Should the computer be used with certain types of individuals? If so, whom? High ability, low ability, introverts,

---

* Total Computer Instruction

adults, basic skills, etc.

Which instructional strategies are most effective on the computer and with which students?

Can the computer be used as a replacement in some subject areas now or do computers need more "intelligence?"

What other characteristics should we be dealing with in an attempt to approve or disapprove of total instruction being delivered via computer?

"Computers and the Learning Society," a June, 1978, report by the subcommittee on Domestic and International Scientific Planning, Analysis and Cooperation, outlines the problems, needs and inhibiting factors of CAI. The report supports the need for research of the psychological effects on individuals who learn from computers and for the development of "test centers" where functions of past work assessment, adaptation/development, implementation and evaluation of more effective CAI hardware, software and courseware are performed.

Assuming that computers (micros, even Plato, etc.) will decrease in cost to an affordable price and assuming that society is not willing to accept computers as a "necessary" educational medium simply because it provides immediate feedback, allows the teacher to work with other students and all the other nice features, when it cannot increase the student's achievement level, the question becomes "What do we need to do with CAI to increase student achievement?"

In an effort to determine whether using the computer as the total instructional experience can produce achievement, we propose a breakdown of the definition for achievement. By doing this, we hope to be able to separate and stress those factors which attribute to greater achievement success on the computer.

Please provide your experience and expertise by indicating whether: I. the items listed affect achievement regardless of medium used, II. the items listed affect achievement when using the computer as the total medium, and III. rank items in order of affect. Add characteristics which you may have found are important.

I. Which items seem to be important aspects, in your experience, of achievement in general?

	Yes (have documented evidence)	No	Don't know	Would like to find out
1. Personality				
2. Beginning educational grade level equivalency				
3. Motivation/interest level towards learning				
4. Success in learning				
5. Self-esteem				
6. Learning strategy imposed by student, teacher or author including sequence of activities/concepts				
7. The instruction meets the exact need (subject area, concept, level of Bloom's taxonomy appropriate for student)				
8. Variety in format of instruction				
9. Motivation/interest level of teacher				

10. Which other characteristics are factors in increasing achievement?

____

____

____

____

____

II. Which items seem to be important aspects of achievement when using the computer courseware for the total instruction?

	Yes (have documented evidence)	No	Don't know	Would like to know
1. Personality				
2. Beginning grade level equivalency				
3. Motivation/interest level of student				
4. Success in learning				
5. Self-esteem				
6. Learning strategy imposed by computer (adaptive, linear, branching, learner controlled, "intelligence" and the sequence imposed)				
7. Instruction meets the exact need				
8. Variety in format				
9. Motivation/interest of teacher				
10. Role the computer plays .. drill .. tutorial .. inquiry .. problem solving .. simulation .. game				
11. Subject area				
12. Familiarity with the computer				

13. Other characteristics

____

____

1 The Effectiveness of Alternative Instructional Media: A Survey, Jamison, et. al., RER, 1974.

2 Selective Review of the Results of Research on the Use of Audiovisual Media to Teach Adults, Campeau, P. L., AVER, 1974.

3 Evaluation Studies of CCC Elementary-School Curriculums, Macken and Suppes, CCC, Palo Alto, California, 1976.

4 Minnesota Corrections Computer Project Evaluation, MECC, 1977.

5 The Effectiveness of CAI, University of Iowa, 1974.

6 Computers and the Learning Society: A Report by the Subcommittee on Domestic and International Scientic Planning, Analysis and Cooperation, Superintendent of Documents, U.S. Government Printing, June, 1978.

III. Rank the characteristics according to what effect (you feel) they have upon the achievement level of the learner.

Your Ranking

1. Personality ____________
2. Beginning grade level equivalency ____________
3. Motivation/interest level of student ____________
4. Success in learning ____________
5. Self-esteem ____________
6. Learning strategy imposed by computer (adaptive, linear, branching, learner controlled, "intelligence" and the sequence imposed) ____________
7. Instruction meets the exact need ____________
8. Variety in format ____________
9. Motivation/interest of teacher ____________
10. Role the computer plays
    .. drill ____________
    .. tutorial ____________
    .. inquiry ____________
    .. problem solving ____________
    .. simulation ____________
    .. game ____________
11. Subject area ____________
12. Familiarity with the computer ____________
13. Other characteristics

____

____

Following the 1st round of questioning, all questions will be analyzed and a new, more specific set of questions will be provided. Please return your reactions to these items by December 1, 1978.

CAI at the University of Akron:
Course Development

A Paper Presented at the 1979 ADCIS Conference
at San Diego, California

Thomas D. Warner
The University of Akron

INTRODUCTION

Over the last decade, the University of Akron has experienced rapid growth in computer usage. One of the out-growths of the University's increased computer usage was the Computer-Assisted Instruction (CAI) Center. Since its creation in 1972, the CAI Center has achieved national and international recognition for its achievements, especially in the area of course development.

The University's method for achieving such success can best be communicated by means of its model for the production of CAI materials. The model presented in this paper was developed and is utilized by this author. Topics included in the model are: formation of an authoring team, specification of behavioral objectives, instructional strategies, media selection, programming and evaluation/revision cycles.

Behavioral objectives should reflect learner characteristics and curriculum design. Instructional strategies should be based upon learning theories and results of research in such areas as frequency of interaction, types of feedback, motivational factors, content variables, and the effects of frequent reviews. It is important to realize that nearly all creative work requires an iterative process of evaluation and revision.

Before such a model can be implemented, there must be a high degree of computer literacy among teachers. This literacy will enable the teacher to accept a shift in his role, making his job more challenging and professional. Such literacy will facilitate the use and development of CAI. Without careful planning and thorough investigations today, future computer-based education (CBE) will consist of scattered and questionable, if not ineffective, approaches to teaching and training. Educators must become involved with CBE so that its development conforms to their dictates, rather than educators conforming to CBE's dictates as formulated by technicians.

NOTE: Purely for convenience, only masculine pronouns are used in this paper.

CAI AT THE UNIVERSITY OF AKRON

A. The University

In 1870, Buchtel College was built amidst a thriving industrial city of 10,000 situated at the summit of the Ohio Canal. By 1913 the College had grown into The Municipal University of Akron. Proclaimed a state-assisted institution in 1967, The University of Akron presently is a comprehensive doctoral degree-granting institution with a student population of over 27,000 and a faculty and staff of 2,500.

B. CAI Implementation

While at Pennsylvania State University, John Hirschbuhl worked with IBM's 1500 CAI system which had outstanding graphic capabilities, audio support, and color slide transport capabilities. In the early 70's he helped create the CARE project, reported by Cartwright, Cartwright, and Robine (1972), which consisted of a self-contained college level course disseminated by a mobile CAI unit that could be driven to teachers who requested the program. It is this type of innovative experience coupled with several years of teaching experience that Dr. Hirschbuhl brought to the University of Akron in 1971.

Hired into the College of Education, John was asked by the new president, Dominic Guzzetta, to investigate the possibility of CAI at the University of Akron. John set up a field trip to Pennsylvania State University for twenty-five enthusiastic faculty members. Upon returning, the College of Engineering hosted a CAI seminar, at which Dr. G. Ronald Christopher of the Ohio State University's CAI Center demonstrated IBM's Coursewriter III interactive language.

Fortified by a concrete strategy for operation, equipment specifications which fit into our existing time-sharing system, and a "cost avoidance" argument, The Faculty Committee on Computer Utilization achieved a level of cost justification that cleared the way for immediate implementation of CAI using Coursewriter III. In 1972, assisted by Lawrence Stolurow and using five teletype terminals, we offered our first Annual Summer CAI Workshop to achieve computer literacy among our faculty, students, and community.

In summary, Dr. Hirschbuhl used his valuable experience to set up field trips and seminars which generated both faculty and administrative support. A research committee was formed which produced a report that spelled out a comprehensive operational strategy, equipment needs and cost justifications for the implementation of CAI. Courseware was obtained from other Universities (mathematics and chemistry) and purchased from the Computer Curriculum Corporation (reading and English).

C. Growth and Development

Computer Center

The University of Akron obtained its first computer in 1964. Within ten years our computing power grew tenfold. Today we have an IBM 370/158 Model III with an attached processor and a PDP 11/40 minicomputer. The new IBM 3031 is on order, to be added to our computing power. All of our 100 interactive terminals can utilize any of the on-line systems (CICS, IIS, FORTRAN and APL). Four PLATO terminals are being used by the Music Department.

CAI Center

The CAI Center has grown to a full-time staff of seven and numerous student assistants. The Center has sponsored seven successful CAI Summer Workshops which have drawn national and international participants, a couple of metric workshops, and an industrial training workshop. We have given workshops, presentations and exhibitions at six National Council of Teachers of Mathematics and at several ADCIS Conferences. Hundreds of prominent national and international guests have toured our Center.

The Human Resources Research Organization identified the University of Akron as an Exemplar Institution in Academic Computing in the categories of student accomplishments and outreach to community and other institutions. IBM has published an application brief, *CAI at the University of Akron* (1976), and is presently distributing five of our CAI courses as Installed User Programs. In short, the University of Akron has achieved worldwide recognition for its accomplishments in CAI.

Community Outreach

Computing services are extended to area hospitals, industrial sites, and to the city of Akron itself. Several area universities and colleges have been hooked up to our computer. Terminals are located in nearly a dozen area schools as well as the Summit County Juvenile Court. At the Court we are researching the possibility of reducing recidivism through the use of a CAI program utilizing behavioral modification and operant conditioning techniques (Warner, 1977). In the CAI Reading Program at Woodridge Junior High School, gains of two years for every school year, as measured by the California Reading test, were consistently experienced for over five years (Katherine Hirschbuhl, 1977). Significant gains in the participants' self-esteem were also documented (Patric McGown, 1977). Similar gains are being achieved in other Akron area schools in both reading and mathematics (Bowman, 1978) and (Konet, 1978), and at the University in many diverse subject areas.

D. Requirements for Implementation

Starting Up

Dr. John J. Hirschbuhl has written an excellent paper which describes the rationale for CAI utilization at the university level, the requirements for implementation, and the cost effectiveness of CAI (Hirschbuhl, 1976). The following are some things to do and not to do in starting up a CAI Center.

Do:

1. Hire an experienced educator who has a CAI background.
2. Solicit and coordinate faculty and administrative support.
3. Plan field trips for "hands-on" experiences.
4. Form a committee to research current literature and attend organizational meetings dedicated to computer based education.
5. Plan and document your strategy for the first few years of operation, addressing all variables.
6. Start small (5 terminals or less), define a few areas of greatest need.
7. Promote staff development and computer literacy among the faculty through seminars and workshops.
8. Procure courseware; buy and/or revise rather than re-create.

Don't:

1. Try to replace faculty.
2. Expect instant success.
3. Fail to collect data on student achievement and use.
4. Fail to obtain the "right equipment" for your situation and budget.
5. Stop trying to secure faculty and administrative support.

Computer Literacy

The differences between critics and proponents of CAI result not so much from variance in basic philosophy as they do from different comprehensions about the nature and capabilities of the computer. A first step toward computer literacy is awareness. "The key to effective use of computer assisted instruction is orientation. The faculty and students need guidance so that problems are minimized and benefits are maximized." (Luce, 1976) Some methods of orientation are: distribution of printed materials, guided tours (and hands-on experience), lectures, seminars, workshops and tape-slide presentations.

A problem of society is the progressive complexity of the problems with which we have to deal. Automation must relieve us of mundane tasks so that we may attack more human problems. A methodical growth of CAI using teacher peers to dissimenate on-going results helps to avoid the unrealistic fear of job loss. Teachers must be intimately involved in order to facilitate their change from the role of purveyor of information to master of resources. The teacher is our basic educational resource and CAI can and should be a tireless colleague, whether it be in our schools or in training programs in industry and the military.

Staff Development

Perhaps the most difficult aspect of managing a CAI Center is the problem of training new and retaining experienced course authors. A prospective author must have teaching experience and believe that all people can learn, but do so at different rates and ways dependent upon mode of presentation, past conditions and aptitude.

The effectiveness of CAI is dependent upon the quality and reliability of hardware, software, and most importantly, the courseware. Faculty have little spare time and therefore not only do they need tangible incentives, such as release time, publication, and royalties, but they also need assistance and guidance in the development of CAI courses. The next section describes a successful tried and tested model for course development.

COURSE AUTHORING

A. Team Approach

CAI production could be compared to giving an amateur sportscaster a radio station and asking him to single handedly produce a days programming. It would indeed be a rare case in which no help would be required from professionals in the field. Potential authors should be interviewed to establish grounds to support motivation and to eliminate unrealistic expectations.

The instructional development team should have expertise in:

1. The content area.
2. Principles of instructional design.
3. Computer programming skills.

The teacher or group of teachers who are the content experts should have knowledge of learning theories, instructional design principles, and at least a rudimentary understanding of programming in order to facilitate communication among team members. Extensive direct interaction with students is also necessary in order to know empirically what kinds of difficulties students are likely to encounter.

"A team should be developed for each academic area in which content is to be developed and should include at least three faculty, a programmer, and an instructional designer." (Dimas, 1978) This is an ideal situation. In many cases, the team could consist of a couple of teachers and an experienced instructional programmer, who is versed in both design and programming. If several faculty can serve as content experts, it is possible to develop courseware in areas where there is mutual agreement as to content and pedagogy, which increases the transfer potential. The instructional programmer should be the team leader, identifying alternative pedagogies, strategies, presentation styles and evaluation techniques.

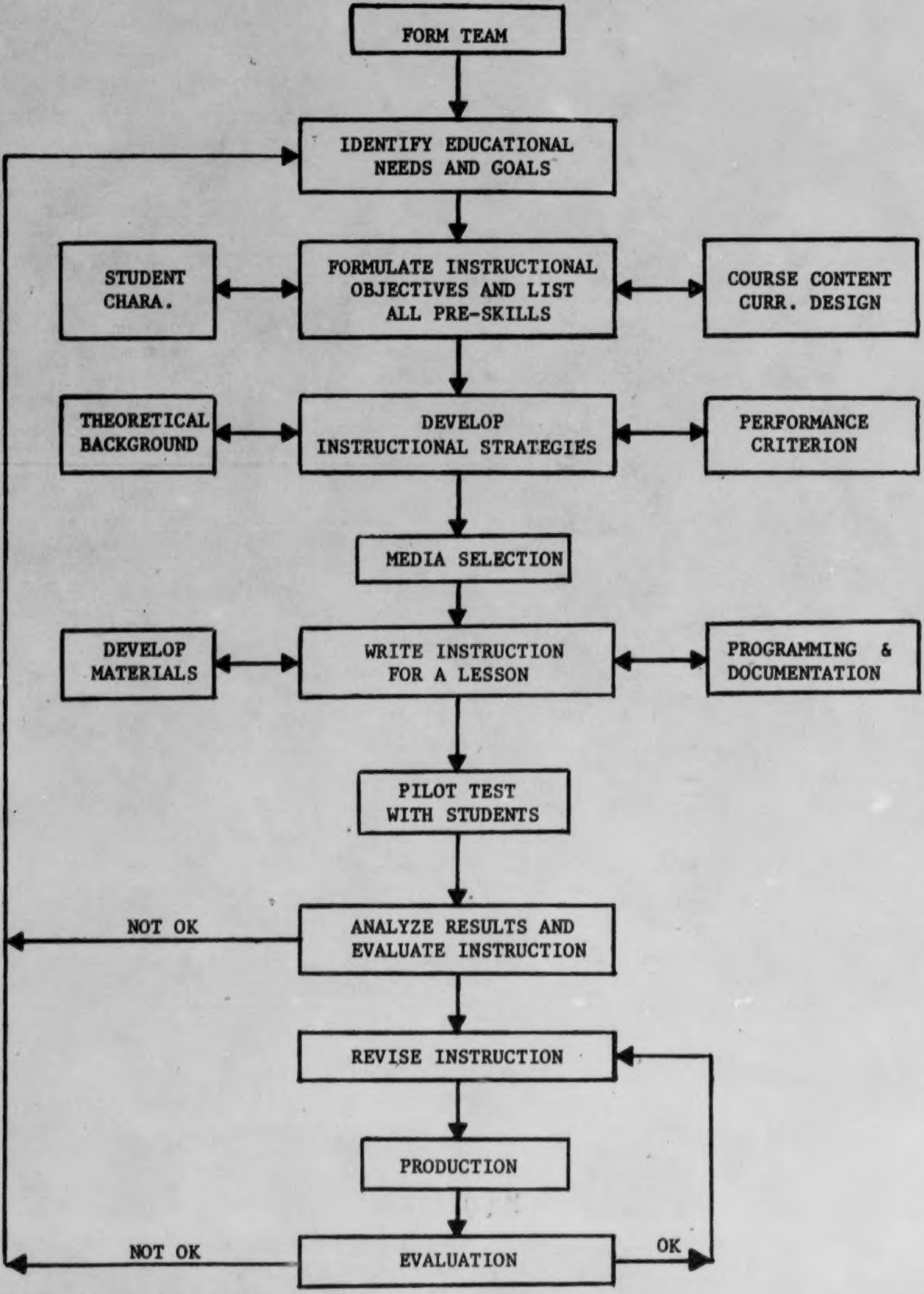

FORM TEAM
IDENTIFY EDUCATIONAL NEEDS AND GOALS
STUDENT CHARA.
FORMULATE INSTRUCTIONAL OBJECTIVES AND LIST ALL PRE-SKILLS
COURSE CONTENT CURR. DESIGN
THEORETICAL BACKGROUND
DEVELOP INSTRUCTIONAL STRATEGIES
PERFORMANCE CRITERION
MEDIA SELECTION
DEVELOP MATERIALS
WRITE INSTRUCTION FOR A LESSON
PROGRAMMING & DOCUMENTATION
PILOT TEST WITH STUDENTS
NOT OK
ANALYZE RESULTS AND EVALUATE INSTRUCTION
REVISE INSTRUCTION
PRODUCTION
NOT OK
EVALUATION
OK

FIGURE 1: A MODEL FOR THE PRODUCTION OF A CAI COURSE

B. Use the Experience of Others

It is too easy for us to submerge ourselves in activities of the present and discount the past as irrelevant, rather than gaining the insight that history offers. Any developer must keep abreast with current research and literature, becoming familiar with previous teaching materials in the area, both computer and other media, so that the experience of others can be employed. It is better to borrow and modify rather than reinvent.

Hands-on experience with someone else's program is essential for one's background. The more exposure the better, for, ideas stimulate creativity. See Karl Zinn (1978) for a list of several organizations and periodicals which promote Computer Based Education. After the content area expert has had extensive teaching experience and student interaction and has sampled the works of others, he is ready to develop his own short instructional module covering an objective. Start small.

C. Model

The model seen in figure 1 depicts the relationship of the elements required in the production of a CAI course. Such a working model is essential in course authoring not only to give you a perspective of when to perform what tasks, but also to help you to include all necessary variables and their interaction before attempting development.

As stated earlier, the best approach is to form a team composed of content experts and an instructional programmer, who has knowledge of both instructional design and programming. The next step is to list your educational goals and to perform a needs assessment in order to identify exactly what you want to do. What follows is a brief description of all of the major steps required for CAI course development, as shown in figure 1.

Instructional Objectives

Educational goals must be translated into instructional objectives which can be learned. An instructional objective should explicitly state what measurable student performances are expected in order to demonstrate proficiency in a learning task under a given set of conditions. In order for an objective to specify the skills or abilities which are required by a particular content area, three parts are necessary.

- * State what behavior the student should exhibit.
- * Specify the given conditions.
- * State the criterion of evaluation.

Example: Find the correct slope of a line given a linear equation, 4 out of 5 times.

Well developed instructional objectives make it easier to determine prerequisites, determine an overall instructional strategy, and write the course content. Before these objectives can be written, you must thoroughly examine two topics, student characteristics and curriculum design.

1. Student Characteristics

Some student characteristics to consider are: maturational factors (especially pre adolescent); prerequisite skills; group characteristics (handicapped, developmental, etc.); and motivational needs. Advanced computer capabilities have not alleviated the need for artistry and creativity in the development of effective and motivating instructional experiences.

The Association for Supervision and Curriculum Development's publication, Humanistic Education (1978), states that effective and humanistic education must be oriented toward student self-concepts. The Coleman Report (1966) found that locus of control was a major determinant of school achievement. CAI promotes more realistic student attitudes which are based more closely on the student's achievement.

CAI has shown convincing evidence that it is an effective medium for building internal control in the student who believes that his life is controlled by external factors. Some contributing factors are: immediate feedback, positive reinforcement, lack of external variables to which a student might attribute his success or failure, individualized interaction, achievement summaries, learner control and adaptive strategies.

2. Curriculum Design

Curriculum Design refers to the analysis and characterization of different subject matters, relating the particular structure of one subject as well as the general variables involved, such as different levels, step sizes, inherent logics, and sequences, hierarchial organization and, in general, sequencing of content elements. (Kearsley, 1978) Content should be so arranged so as to accentuate both the difference between related concepts and the scope of each concept. The content experts of the team must get together and agree upon the curriculum design.

Instructional Strategies

1. Theoretical Background

Principles, well supported by the evidence of research, constitute a good basis for making decisions. Your development method should adhere to methods of science rather than trial-and-error. An instructional strategy can be defined as a plan of relatively structured procedures by which specific objectives are achieved. "Instructional design is closely related to teaching and learning but it differs from these in that it is systematic, empirical and prescriptive in nature." (Kearsley, 1977).

A model of teaching must be *descriptive* if teachers are going to use it. It must be *prescriptive* if it is going to be used for decision-making. It also must be *response-sensitive*, if it is adaptive. The relationship between a model of teaching and a model of learning is not one-to-one. Different models (and different instructional strategies) might be developed to account for the same set of dependent variables used to measure learning.

The heart of any CAI program lies in its instructional strategies which must be based upon a well founded theory of learning. Developmental theories (Piaget, Bruner, Erikson, Skinner, Sears, Kolberg, etc.) all suggest that certain skills and abilities are age-related. Therefore, learners should not be expected to successfully cope with abstract verbalization of concepts prior to experience with their concrete counterparts.

General learning strategies should be established. A TICCIT study showed that "the effectiveness of a particular learning strategy is independent of subject matter" (Jones, 1978). In the area of instructional research, the computer can provide more refined information as to how students learn.

2. Criterion Measures

Your adaptive strategy, and thus an individual's path through your course, must rest on the attainment of certain criteria that you set up. Appropriate instructional alternatives can be determined by considering such things as:

* Whether the student covers many steps at once or singly.
* Whether the student is slow and accurate or fast and inaccurate etc.
* What the student's performance was on similar topics or prerequisite material in this and other sections, units or courses.

Important information about student performance may be derived from the frequency and sequence of correct, incorrect, and unanticipated answers; error analysis determined by what part of the problem was missed most; the expressed confidence of the responses; the latency of response; and the sequence of steps used in problem solution.

3. Modes of Instruction

Some modes of instruction are: problem solving, drill-and-practice, tutorial, gaming and simulation, data analysis, and model building. Most CAI courses are composed of an integration of several of these modes or instructional strategies.

One of the problems of CAI is the translation of generally vague instructional practices and learning theory into computer programs which are quite precise. Various instructional strategies (modes) and sequences are appropriate for different categories of learning. For example, the tutorial mode corresponds to expository and deductive learning; simulations correspond to inductive learning, etc.

Techniques of prompting, shaping and fading have slightly different applications and meanings for the various modes of instruction. No single strategy or sequence is suitable for all applications.

A good CAI course individualizes instruction by segmenting instructional sequences and providing an instructional strategy that is appropriate to the talents, interests and background of the learner. You must try to determine what processes or procedures are necessary to teach certain types of learners and tasks by examining the full range of possibilities each strategy offers. You should be prepared to address all possible weaknesses that students are likely to exhibit.

Adaptive mastery learning models readily adjust to the difficulty of the objective. An adaptive model is the best way to accommodate individual differences. Together with adaptive programming control, a high degree of learner control is favorable, if not essential, beyond the elementary school level.

Do not use the computer like a textbook or lecture! Interaction is a major quality of CAI. You cannot use interaction too plentifully. The student must be continually involved with frequent questions. A CAI teaching strategy consists of presenting content in small and properly ordered segments that can be mastered one at a time, prompting correct responses or correcting incorrect ones by feedback and using remedial loops, i.e., reordering the material to be learned. Some essential elements in authoring are: immediate informative feedback, positive reinforcement, repetition of concepts with frequent reviews and keeping the learner informed of his progress and achievement through report summaries.

Media Selection

Media selection is choosing a delivery system for the presentation of learning objectives. A pluralist philosophy of media use should be adopted. The computer element should be integrated into the teacher's lectures and use of textual and video materials. Multi-media use can be under control of the computer. However, the medium must not compete for meaning with the instruction. See Rita Wagstaff (1978) for a discussion of media selection models proposed by Allen; Tosti and Ball; Boucher, Goltlieb, and Morganlander; Kemp; and Romiszowski.

The type of computing service needed must be determined. Then the selection of the kind of terminals that you want to use can be made. Possessing a reliable computer system is very important in maintaining student motivation and curbing any unnecessary frustration. A review of computer equipment and CAI terminal characteristics can be found in Braun (1978) and Bork (1978).

Write Instruction

If the production team has searched current literature, had "hands-on" experience with others' programs, properly constructed their instructional objectives and specified an instructional design taking student characteristics, terminal capabilities and the content area into consideration, the writing of the instruction should be easy. Normally, over 70% of course authoring is in the design, which is formulated before the actual writing of content and programming.

Courses produced to meet local needs are not accepted by other schools as easily as courses cooperatively developed. A flexible modular format will increase the potential for transfer. A flowchart of the instruction flow is necessary. The use of text editors and system functions are time savers, however, any special programming functions used should be carefully documented.

Evaluation

Development, as with all creative work, requires an iterative process. The first test of your material is to find out what has to be revised not if a revision is necessary. The courseware should be field tested with a small number of students and revisions should be made as indicated by student performance information. On-line data reflects very well any changes in a lesson and is a good tool for evaluation of lesson revisions (Weaver, 1978). Many evaluation/revision cycles will be necessary.

Some causes of failure are:

1. All pre-skills were not tested in the pretest.
2. The instruction does not adequately cover or relate to the objectives.
3. Faulty design or inadequate criteria.
4. The post test does not correspond to the instructional objectives.

Generally, it is the lesson that fails, not the student.

D. Style

Concerning the content of your course:

1. Be logical, concise and clear.
2. Use simple vocabulary.
3. Make key points visible (aerate your material).
4. Present only small amounts of information at a time.
5. Use many examples, get the reader to participate.
6. Do not use slang.
7. Never use negative or derogatory feedback.
8. Program the beginning of each lesson to guarantee initial success.
9. Allow a high degree of learner control.

Concerning the programming:

1. Use a modular or systems approach.
2. Use spacing to highlight your structure.
3. Make use of looping, reusing code that is frequently called.
4. Document everything in as much detail as possible.

Some parts of the documentation should be written for the teacher, some for the computer operator, some for the system programmer, and some for the student so that he can use the course effectively. Therefore, both an

Instructor's Guide and a Student Guide must be written. For an excellent source of what to include in these Guides refer to CONDUIT's Author's Guide (1978). This publication is very well written and organized. Any perspective author should read it.

CONCLUSION

Both the development and utilization of CAI courses at the University of Akron are increasing rapidly each year. As seen at the University of Akron, specific requirements for implementation of CAI should be followed in order to assure success. One of the first steps is to achieve computer literacy among administrators and educators. This literacy must be obtained before educators will assume the role changes necessary for the effective use and development of CAI.

Figure 1 represents a successfully used model for CAI course production. A development team should possess expertise in (1) the content area, (2) instructional design, and (3) programming. Principles, well supported by the evidence of research, constitute a good basis for making decisions. The experience of others as well as current research on learning theories and learner characteristics should be used when formulating the instructional design and selecting and integrating various media. No single strategy or sequence is suitable for all applications. Be creative!

Your programming style and documentation are very important. Use clear and concise statements and present only small amounts of material at a time. Your course should provide for a high degree of learner control and be continually interactive, providing informative feedback and positive reinforcement.

Computers will be in most homes within the next decade. The crucial limiting factor for the future is not computers, but people. The major ingredients that will affect the growth of CAI are the professional development of sophisticated course authors, the implementation of videodisc systems, and the foresight and dedication of today's CAI implementors. Ultimately, a cybernated society should be able to produce more intellectually and emotionally competent people.

References

The Association for Supervision and Curriculum Development. Humanistic Education: Objectives and Assessment. Washington D.C., 1978.

Bork, Alfred. "Machines for Computer-Assisted Learning." Educational Technology. Vol. XVIII, No. 4. April, 1978.

Bowman, Burdene K.. "Results of Computer-Assisted Instruction at Bath Elementary School." Unpublished Paper, 1978.

Braun, Peter. "CAI Terminal Characteristics." New Directions in Educational Computing. ADCIS Proceedings. Dallas, Texas: March, 1978.

Cartwright, C. A.; Cartwright, G. P.; and Robine, G. E.. "CAI Course in the Early Identification of Handicapped Children." Exceptional Children, 1972.

Coleman, J. S.; Campbell, E. O.; Hobson, C. J.; McPortland, J.; Mood, A. M.; Weifield, F. D.; and York, R. T.. "Equality of Educational Opportunity." Washington, D. C.: U. S. Office of Education, 1966. Catalog #FS5. 238-38001.

Computer Assisted Instruction at the University of Akron. Applications Brief. International Business Machines Corporation: White Plains, New York, 1976.

Dimas, Chris. "A Strategy for Developing CAI." Educational Technology. Vol. XVIII, No. 4. April, 1978.

The Faculty Committee on Computer Utilization. "Position Paper on Computer Assisted Instruction for the University of Akron." University of Akron: December, 1971.

Hirschbuhl, John J.. Blueprint for the Future of Computer Based Instruction. The University of Akron, 1976.

Hirschbuhl, John J.. "CAI in the University Environment." The University of Akron, 1976.

Hirschbuhl, Katherine. "The Teacher and Technology: The Woodridge Study of CAI, A Three Year Study." ADCIS Proceedings. The National Research Council: Washington D. C., 1977.

Jones, Michael C.. "TICCIT Applications in Higher Education: Evaluation Results." *New Directions in Educational Computing*. ADCIS Proceedings. Dallas, Texas: March, 1978.

Kearsley, Greg P.. *Instructional Design in CAI*. Bethesda, Md.: ERIC Document Reproduction Service, ED 152 293, 1977.

Konet, Rose Marie and Hirschbuhl, John J.. "Final Report For 1977-78 CAI Reading Project, Akron Public Schools. The University of Akron: 1978.

Luce, Carol. "Orientation: The Key to Effective Use of CAI." Paper presented at the Jan., 1976 meeting of ADCIS in Santa Barbara, California.

McGown, Patric. A letter to Dr. James Ervin, Assistant Superintendent — Curriculum and Instruction. Akron Public Schools. March, 1977.

Peters, Harold J. and Johnson, James W.. *Author's Guide*. CONDUIT (The University of Iowa), 1978.

Wagstaff, Rita. "Media Selection: Alternative Delivery Systems." *New Directions in Educational Computing*. ADCIS Proceedings. Dallas, Texas: March, 1978.

Warner, Thomas D.. "An Evaluation of the CAI LEAVE Program at the Summit County Juvenile Court." The University of Akron, 1977.

Warner, Thomas D.. "Authoring a CAI Mathematics Course." Master's paper. The University of Akron, 1976.

Weaver, Tomar A.. "On-line Data in CBE." *New Directions in Educational Computing*. ADCIS Proceedings. Dallas, Texas: March, 1978.

Zinn, Karl L.. "Sources of Information About Computing in Instruction." *Educational Technology*. Vol. XVIII, No.4. April, 1978.

OUTSTANDING LARGE-SCALE PROJECTS IN BUSINESS & EDUCATION

Harold F. Rahmlow, Ph.D.
The American College
Bryn Mawr, PA 19010

Computer-based education is in its early stage of development when contrasted with other mediated forms of education. Because of this newness, many educators and trainers are unaware of the significant applications that have been made of the technology. The purpose of this paper is to briefly illustrate three outstanding large-scale computer-based projects. A diversity of applications is emphasized in order to demonstrate the versatility of computer-based education.

The material for this paper was derived from an intensive study of computer-based education which had as its emphasis an investigation of the potential for computer-based education in the insurance business and in non-traditional higher education. Computer-based instruction projects which appeared to have potential were identified by hardware and software vendors, through an analysis of the literature, and through discussions with other experts in the field.

In the spring of 1978 the author personally visited numerous projects including the three described in this paper. Many outstanding projects were noted. Three have been selected to demonstrate the diversity of computer-based instructional applications: CICERO, a computer-managed instruction project operated by the Open University of Great Britain, a teller training project operated by Western Bank Corporation, and a flight training program operated by American Airlines.

The Open University

The Open University of Great Britain was established in 1969 to serve adults throughout Great Britain. It is a non-residency institution with students studying in their home environment. The University makes use of printed materials, radio, television, and computer technology to deliver instruction. Course development teams and administrative functions are housed at Walton Hall, about an hours journey north of London.

CICERO, one of the Open University's computer-based projects is utilized in three areas. One, as a testing mechanism to pinpoint areas of student deficiency in knowledge and understanding; two, to provide brief explanations and follow-up exercises in identified areas of difficulty; and three, to suggest the time management strategies for future study.

Prior to the introduction of CICERO, students received assessment through tutor-marked assignments, computer-marked assignments, and through questions embedded in a correspondence text. The tutor-marked assignments are much like standard essay questions in traditional universities. They are marked by the students tutor and once a grade has been recorded on the student's record are returned complete with written comments. However, there is little opportunity for face to face discussion of the work and the success of the method of teaching depends largely on the skill of the tutor in making comments as succinct and helpful as possible. Computer-marked assignments, on the other hand, take the form of long lists of multiple choice questions. These are sent to the University and processed. Since the resulting grade is used for assessment, no feedback can be given before a specified cutoff date which usually coincides with the end of a particular work period on a certain portion of the course. When the feedback arrives it takes the form of a letter grade indicating overall progress. It can be seen, therefore, that these two methods of assessment do not provide opportunity for real assistance in learning. The third method, text embedded tests, are considered to be limited in scope and quite general in the advice provided. CICERO attempts to fulfill the need for diagnostic evaluation.

CICERO can be used either on-line or off-line through the postal service. The Open University provides a wide network of computer terminals which are distributed throughout the country and are available to most students. CICERO operates on the DEC 20 series of hardware and is programmed in BASIC+2.

Students are mailed questions at the beginning of their work block and are advised to attempt to answer them part way through the allocated time. Using an on-line terminal, CICERO prompts for the answers to the questions, provides annotated answers, and if necessary indicates remedial paths.

Students using the postal service send their forms to the University where they are processed and returned by post. In the first year of operation approximately 65% of the students chose to use the on-line option, and the remainder the post.

Although a number of features of CICERO contribute to its success, the outstanding contribution of this project is to provide high quality personalized feedback to students who are participating in distance education.

Additional information on CICERO can be obtained from:

Dr. Aldwyn J. R. Cooper
The Open University, Walton Hall
Milton Keynes MK7 6AA
ENGLAND

Western Bancorporation

Western Bancorporation (WBC) is a bank-holding company which owns major interest in 22 locally managed and operated commercial banks in 11 western states. Ranked according to assets, it is the tenth largest U.S. bank-holding company with the United California Bank of Los Angeles as its anchor bank. WBC is in the process of implementing TIPS (Teller Item Processing System). When TIPS is completely installed in 1978, its communications setup will include the country's largest on-line banking network.

The challenge facing WBC was to train approximately 6800 tellers and 1500 bank officers in the use of TIPS. The magnitude of the population, its geographic dispersion, and a very tight time line eliminated a number of training options. After evaluating the potential of programmed instruction and computer-based education it was determined that computer-based education was the appropriate media. Computer-based education has proven to be learning, cost, and logistically effective in the short time frame available for implementing training. Personnel receive computer-based education using the IBM 3604 financial system terminal. This terminal is the device which is used by tellers in their day to day work.

The computer-based training is enabling WBC to effectively train large numbers of personnel in diverse geographic locations rapidly and effectively. The computer-based education system is implemented under Boeing's Scholar/Teach III and runs on IBM 370/158 systems out of El Segundo, California. Because of the parsimonious structure of Scholar/Teach III, the training is able to be carried out simultaneously with normal production operations without noticeably degrading the production environment.

Additional information on this application can be obtained from:
Michael D. Reis
Vice President
Western Bancorp Data Processing Company
871 South Nash Street
El Segundo, CA 90245

American Airlines

American Airlines is using computer-based training as a cost effective method for flight training for DC-10 flight crews. The training is provided through high quality simulation of cockpit systems.

American selected those areas of its flight training curriculum which were most difficult and which were thought to be most cost effectively implemented in computer-based education. The computer-based material which is integrated with slide tape presentation replaced some personal instruction, and a significant capital investment in flight simulation equipment.

The training is carried out on the CDC PLATO system in American's flight training center at the Dallas-Fort Worth International Airport.

Although the flight training has a number of interesting features the most notable is the extensive use of "state of the art" simulation and graphics on PLATO. In the development of the materials, American received strong support from Control Data experts in Minneapolis.

Additional information on the American project can be obtained from:
Mr. Jerry Plemons
Flight Training Center
Dallas-Fort Worth International Airport
Dallas, TX 76125

A Synthesis

The projects described above can be analyzed from a number of points of view. The analysis reveals a number of central themes as well as individual variations.

Paramount to all projects is the theme that computer-based education has been used to solve a specific education or training problem. The learning problem was first identified and carefully delineated, and subsequently computer-based education was selected as a cost effective means of meeting that need.

Likewise, each project uses computer-based education as part of a broader educational or operating environment. The computer project does not stand in isolation but rather is an intregal, and in the cases of WBC and American Airlines, a critical aspect of the corporations's training strategy.

All three projects are marked by the individualized nature in which they meet student needs. Because each project deals with adults, and two of the three projects operate in an environment in which training time is costly, individualization is of major significance.

The projects also differ in a number of respects indicating a diversity of means for solving learning problems through computer-based methodologys. The Open University and WBC projects operate with a decentralization of learners, while the American Airlines project operates centrally. The Open University project has as its goal the providing of diagnostic and prescriptive assistance to students in meeting course academic goals. The WBC project provides simulated training in a production environment, while the American Airlines project provides a simulation training tool. The effectiveness of the Open University project is judged by the manner in which it contributes to student progress toward academic goals. The WBC and American Airlines projects are evaluated with respect to their capacity to provide a cost effective method of achieving training goals, goals which are critical to each organizations operation. Finally, the computing environment and systems for each is different. The Open University operating on DEC equipment in tandem with other academic and administrative operations. The WBC project operating in tandem with other bank production operations using Scholar/Teach III and the American Airlines operating on PLATO, a computing system designed for and dedicated to educational applications.

These three outstanding uses of computer-based education indicate a diversity of successful application of computer-based technology to meeting educational and training needs. The descriptions of the individual projects have purposely been kept concise. It is not the intent, nor the prerogative, of the author to go into great depth with respect to the features of individual projects. The personnel associated with the projects are much more capable of providing in-depth information.

COMPUTER ASSISTED INSTRUCTION IN AN EVERYDAY SCHOOL ENVIRONMENT

Dr. Deborah A. Cheves, Evaluation Consultant
Dr. Terry E. Parks, K-12 Mathematics Supervisor

Shawnee Mission Public Schools
7235 Antioch
Shawnee Mission, Kansas 66204

INTRODUCTION

The Shawnee Mission Public Schools initiated its first program of computer use for instruction in 1969. This initial effort, directed toward developing student programming capabilities, led to establishment of a three semester computer science offering presently being provided in each of the district's five high schools.

In 1970 the district moved to add a ten terminal APL system to its instructional computing capabilities. Driven by an IBM 360/40 computer, this on-line system was selected to expand services for CAI, computer managed instruction, and student problem solving. Being separate from the computer science program, APL programs were developed by teachers and students on an unpaid basis or were adapted from other APL users. Unlike computer science, the APL system required one non-instructional person to maintain it.

This system was supported and maintained for four years during which much controversy arose regarding cost/benefit relationships. Concerns about program development costs, program ownership rights, access limitations, maintenance expenses, and administration's inability to report clear cut benefits in terms of student learning led the Board of Education to direct administration to discontinue APL as of the close of 1974-75 school year. The Board insisted, however, that instructional computing which could meet the following criteria be retained:

1. directly serve students
2. serve students cost effectively
3. require minimum expense for program development and inservice

With these criteria and a $50,000 budget bids were let to seek an appropriate system. The bid specifications called for providing CAI, computer managed instruction, and student problem solving capabilities. No vendor could meet all requirements within budget, and it was therefore decided that CAI should have precedence. On this basis the bid from Computer Curriculum Corporation for a 10 terminal minicomputer system to serve reading, mathematics, and language arts K-7 was accepted. This system was installed in 5 schools (4 elementary and 1 junior high) in the fall of 1975.

IMPLEMENTATION AND EXPANSION

Evaluation results from the 1975-76 pilot year (see evaluation section) were sufficiently impressive to lead the Board of Education to authorize expansion each year since. Expansion details are outlined in Table I.

Table I. Shawnee Mission CAI Growth

		Costs[1]					
Year	Equipment	Personnel	Equipment	Total	# Students Served	Cost Per Student	# Schools Served
1975-76	1 computer 10 terminals	$ 1,000	$ 41,441	$ 42,441	284[2]	$ 149	5
1976-77	1 computer 30 terminals	$ 1,000	$ 110,753	$111,753	1,206[2]	$ 93	15
1977-78	2 computers 42 terminals	$ 1,000	$ 165,122	$166,122	1,348[3]	$ 123	21
1978-79[5]	2 computers 72 terminals	$18,000	$ 171,154	$189,154	2,310[4]	$ 82	36
1979-80[6]	2 computers 102 terminals	$19,000	$ 245,000	$264,000	3,275[4]	$ 81	51
1980-81[6]	2 computers 128 terminals	$20,000	$ 307,500	$327,500	4,110[4]	$ 80	64

[1] costs reflect purchase except during 1975-76 pilot year which was lease

[2] counts all students served

[3] counts only students served 2 hours or more

[4] estimate based on students served 2 hours or more

[5] based upon approved budgets

[6] projected assuming continuation of proposed plan

At present the Shawnee Mission School District maintains 64 schools all of which will be receiving CAI service in 1980-81. The present (1978-79) school year has service to 29 elementary schools, 5 junior high schools, and 2 senior highs. CAI is being delivered through Title I classrooms, learning centers, and reading specialist programs in the majority of schools. Two schools, however, have a terminal in regular classrooms where mathematics is taught as a departmentalized subject.

Since the initial, pilot year, implementation has been initiated with inservice provided by school district personnel. Teacher training involves one-half day of overview, curriculum explanation, and program description followed by one-half day of introduction to operating procedures such as enrollment, change, and report routines. Each school terminal proctor then receives on going training through day-to-day contact with the CAI Resource Teacher. Cost for inservice averages $45 per teacher.

The CAI Resource Teacher is the person responsible for overseeing system maintenance, communicating with the telephone company, working directly with teacher users, assisting with inservice, and monitoring system usage and effectiveness. Monitoring reports (see evaluation section) are prepared on a periodic basis and forwarded to principals, teachers, and administration. Constant monitoring and careful maintenance tends to keep usage high while maximizing effectiveness.

Implementation has been accomplished utilizing staff who have little or no traditional computer science background. At present neither the CAI administration, resource teacher, nor any users know how to program the computer. All software is leased and the major technical/mechanical requirements are serviced via a maintenance contractor with minor repairs being made by the resource teacher. Down time on the computers averages 6-10 days per year. Turn around on routine minor terminal, modem, etc. services is an average of 2 hours.

By 1981-82 all equipment will have been purchased. After purchase is completed, the maintenance, personnel, telephone, and software costs should average approximately $42 per student per year. Based upon evaluation data from the 1976-77 results this should translate into ~$20 per student year of learning growth in mathematics, ~$35 per student year of growth in reading, and ~ $14 for similar growth in language arts.

CURRICULUM

The major attraction to the Computer Curriculum Corporation CAI offerings lies in their curricula. Systems available from this company include mathematics (for grades 1-6 and 7-8), algebra, language arts (strands and topics for grades 3-6), reading (for grades 3-6 and for comprehension), adult arithmetic skills, critical reading skills, fundamentals of English, problem solving, Spanish mathematics, and GED programs. At present the Shawnee Mission system drives mathematics 1-6 and 7-8, algebra, both language arts subjects, reading 3-6 and comprehension, and adult arithmetic skills.

Each program is designed around an approximate grade equivalent and strand format. For example, the mathematics program contains 14 strands covering grades 1 through 6.9. These strands range from elementary number concepts through operations with integers. The other curricula are similarly organized.

One other feature of these programs is the offering of drill and practice through mixed drill format. This feature is one that, though effective, is difficult for classroom teachers to accommodate in their routine teaching techniques.

These curricula have demonstrated their value through numerous evaluation studies. Likewise, as can be seen in the following section, similar results have been observed in the Shawnee Mission CAI program.

EVALUATION

During the pilot program in 1975-76 284 students used CAI services in five schools. The evaluation design utilized experimental and control groups with standardized test data in mathematics and reading. A standard regression technique applied to the test data yielded the following Student Development Indices:

	CAI	Control
Reading	2.60	1.29
Mathematics-Concepts	4.67	2.03
Mathematics-Problems	4.07	4.50

This study clearly indicated that CAI was effective in reading and mathematics-concepts. Error limits of ± 0.4 precluded determination of significance on mathematics-problems.

During 1976-77 1,206 students were served by the computer in mathematics, reading, and language arts. Students expected to gain 7 months averaged 1 year, 8 months and 1 year, 5 months gain in reading and mathematics respectively according to KeyMath Diagnostic Arithmetic Test and Woodcock Reading Mastery Test.

Additional analysis indicated a significant relationship between time spent working on the computer and KeyMath Diagnostic Arithmetic Test (Figure 1) and CAI grade-placement gain (Figure 2). According to the regression line a student who spent 1500 minutes on mathematics at the terminal would be expected to gain 2.1 years at grade 2 and 2.8 years at grade 6. Expected gain in reading per 1500 minutes was 1.2 years. Language arts growth during a 1500 minute time period was found to exceed 3 years.

Sixteen bilingual students working on the language arts curriculum were particularly responsive. Not only did they work at the terminal for the prescribed amount of time faithfully but their gains ranged as high as 6.5 months per month. Thus, computer usage was considered a very important variable in increasing achievement.

Although a system of process evaluation or monitoring of CAI usage within each building had been used during 1975-76 and 1976-77, additional attention was paid to individual building usage as a result of evaluation findings. A goal of 30 students working 10 minutes daily was set for each building, and a monthly report was prepared to indicate progress toward the goal. The 2 most important pieces of information were minutes per student per day and growth per month. Since students were to spend 10 minutes per day, the number of minutes actually spent enabled monitors to note how many days students were working and to make adjustments to assure greater usage. As reported above, growth and usage are clearly related. When a building's growth and usage did not coincide, recommendations for better instruction were made. At the conclusion of the 1977-78 school year, the average number of minutes per student per day in 15

Figure 1 . Scatterplot of Gain in Mathematics as Measured by CAI Grade Placement Scores and Time Spent in Mathematics CAI Curriculum - Partial Sample.

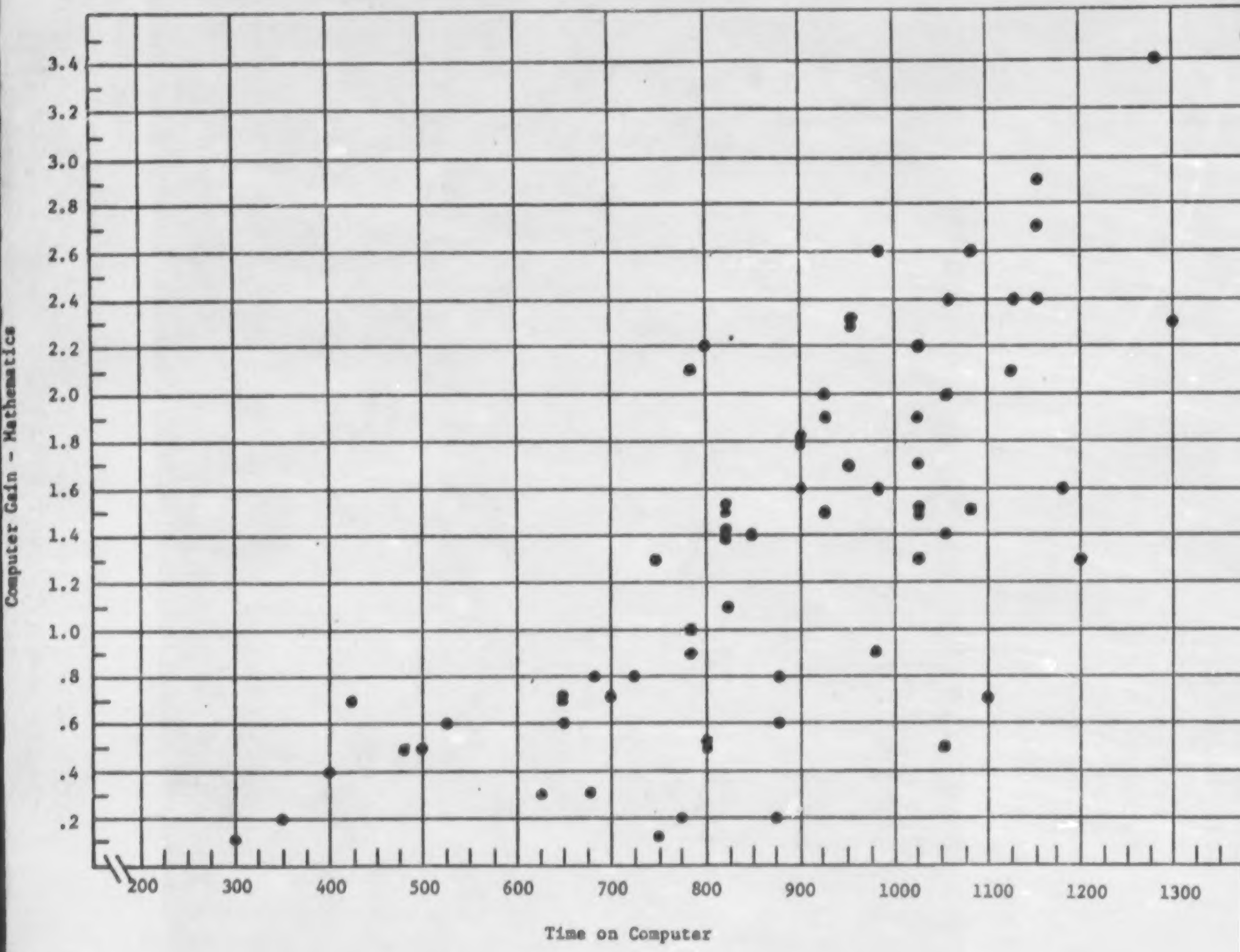
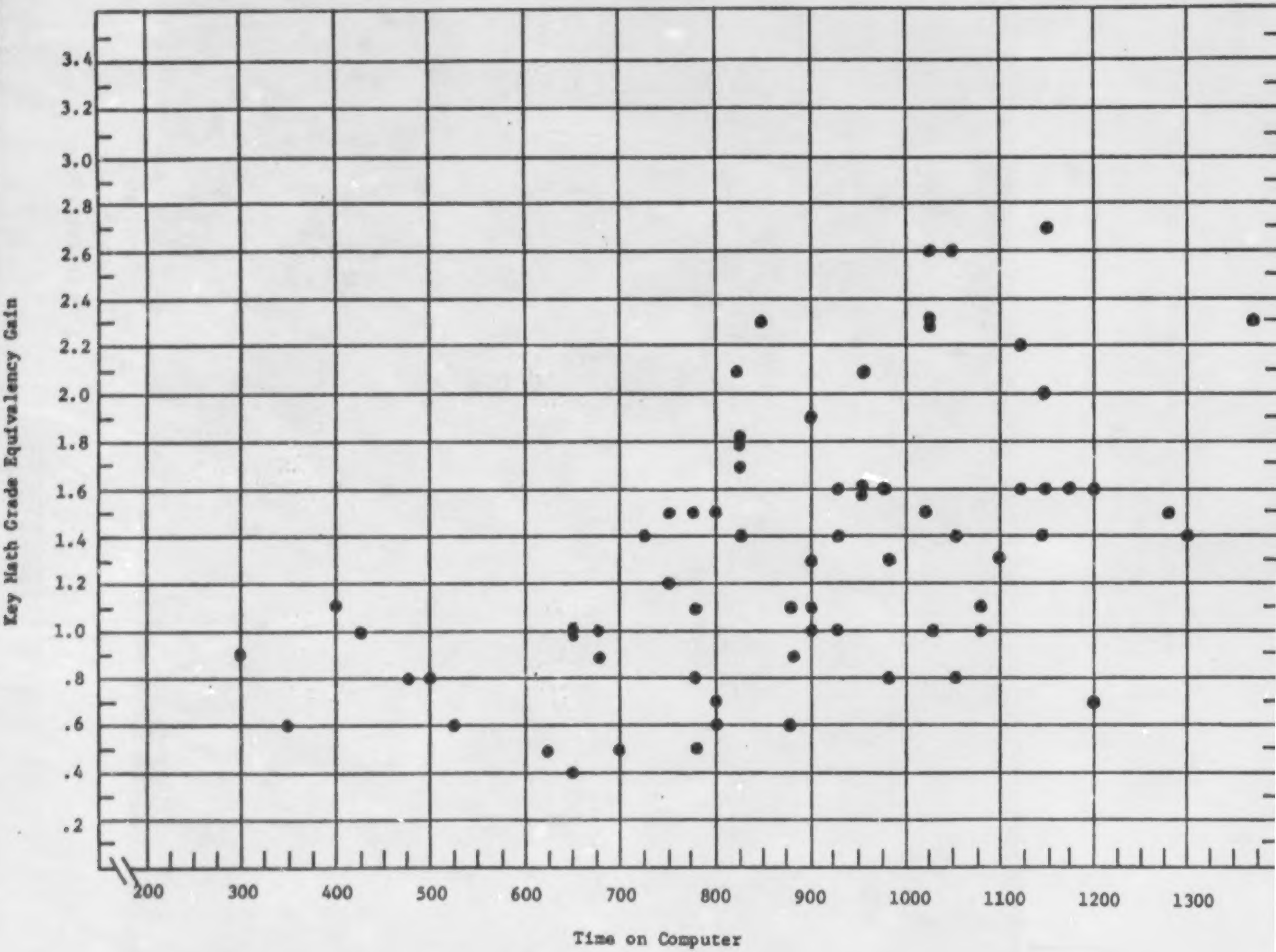


Figure 2. Scatterplot of Gain in Mathematics as Measured by Key Math Diagnostic Arithmetic Test Grade Equivalency Scores and Time Spent on Mathematics CAI Curriculum - Partial Sampl



schools was:

	Minutes/day	Growth/mo.
Mathematics	8.2	2.8
Reading	7.8	1.0
Language Arts	8.1	3.4

Thus, on the average a student worked on the computer 4 out of 5 school days during 1977-78. If this average was persistently low in a building despite monitoring and follow-up recommendations, terminals were removed from the school and placed elsewhere. During 1977-78 computer usage exceeded 15,000 hours whereas 11,168 hours were logged in 1976-77. Monitoring and support has proved to be a very useful tool in implementing a program which requires a rather significant change in viewpoint on the use of students' time.

Student development was studied at the conclusion of 1978 in schools which had terminals for three consecutive years. Those students who had taken an Iowa Test of Basic Skills test at the end of 3rd grade in 1975 and had worked on the computer sometime during the three year period were singled out for study. Thirty students in reading and 33 mathematics students also took the Iowa Test of Basic Skills at the conclusion of their 6th grade year. The following Student Development Indices were found using the same standard regression technique as in the 1975-76 pilot.

Reading	4.95
Mathematics	4.63

Clearly the computer program enabled students to make significant gains during grades 4-6 above what was predicted for these students at the end of 3rd grade. Furthermore, in 1978 students attained greater gains in both reading comprehension and math problems than those observed in the 1975 pilot study. Students in the 1975 and 1978 samples received computer assisted instruction because they had remedial needs. When compared to the district student population which more closely represents a normal distribution, these samples of remedial students equalled or exceeded the student development index attained by those students who had not used the computer.

CONCLUSIONS

The experiences with CAI offer some guides to school districts investigating the potential of such programs. First, few school districts have sophisticated computer equipment and even fewer have well trained computer staff. Consequently, instructional computing which is a far less well developed field than administrative computing, is a relatively untried, insecure yet necessary field for most school systems. The results and experiences detailed here lead us to the following suggestions which we hope can serve others to avoid some of the pitfalls we have found in developing an instructional computing system.

1. CAI can be used in an everyday school environment.
2. Local district computer (including CAI) program development is less efficient and less cost effective than a turn key system.
3. Computer usage for instructional purposes can best be accomplished by:
    a. using a system which focuses on teacher instructional skills rather than technical skills
    b. providing constant monitoring of usage and student learning
    c. minimizing turn around time on service calls.
4. CAI can be cost effective.
5. When used in conjunction with regular classroom and/or remedial instruction, CAI drill and practice shows significant learning growth.

APPLICATION OF BIBLIOGRAPHIC INFORMATION RETRIEVAL TECHNIQUES TO COMPUTER-BASED EDUCATION MATERIALS

John S. Edwards
University of Georgia
Athens, Georgia

Pier Paolo Pisani
GNOSIS, Inc.
Boston, Massachussetts

INTRODUCTION

This study investigated the feasibility of using a bibliographic indexing model for unifying the elements of a computer-based education (CBE) program of the University System of Georgia. The CBE program includes instruction, testing, guidance, and management systems which have been developed locally or acquired from sources outside the University System. As the collection of CBE materials grew to substantial proportions, it became essential to provide some means for coordinating further growth and to provide a means for users to determine the content and to perceive the interrelationships of courseware, test items, competency statements, and other teaching materials of the University System Network.

This report will describe the techniques of bibliographic information retrieval applied to the CBE materials, including the development of indexing tools, the structuring of CBE materials into data element oriented files for the purpose of computer-assisted storing and retrieving, and the computer based system employed to generate indexes. In addition, quantitative measures of recall and precision are reported to evaluate the use of the bibliographic indexing model for managing CBE materials.

INFORMATION RETRIEVAL TECHNIQUES

Vocabulary Control

Information retrieval as an activity is concerned with the manipulation of classes of information. Each element of information in a system must be assigned to a class in such a manner that persons seeking information may recall it by means of the proper classification. Concept indexing is one technique for effecting a useful classification of information. Concept indexing uses terms (words, symbols, etc.) to identify the concept classes. For example, in a collection of CAI lessons in English grammar one would find such concepts as "agreements in case and gender between the subject of a sentence". This concept in turn would be indexed under such classes as "subject", "verb, "noun", "nominative", "dative", "case". This list of words (terms) is then used to index, that is, place labels on, all elements of information which contain the original concept as stated above.

An indexing scheme which uses natural language words suffers from five major problems:

1. The synonym problem in which a concept or similar concepts may be labelled by various words. Elimination of synonyms allows a reduction in file space required for the collection and permits a more precise definition of the concept labels.

2. The homograph problem in which terms are spelled alike but have different meanings. The most common case of this being the confusion between noun and verb forms of words such as "paint" and "bed".

3. The generic search problem in which a word may subsume a linguistically large space, that is, it may contain a large number of sub-concepts. This problem causes a lack of precision in searching since many non-relevant elements of information will be retrieved.

4. The syntactic problem in which false relationships between index terms may be generated when searching. Thus, a request using the terms "Venetian" and "blind" may retrieve information on "venetian blinds" and "blind Venetians".

5. The problem of content in which a word may be viewed from various contexts. For example, the word "wood" may be viewed as a product of the lumber industry, a fuel, or a material for construction.

In order to control these problems, an hierarchical thesaurus was generated according the American National Standards Institute (ANSI) Guidelines for Thesaurus Development. (1974). As recommended by ANSI the following conceptual relationships were established:

1. USE- USE FOR (UF). This relationship is designed to ameliorate the difficulty presented by synonyms and near-synonyms. As with all the ANSI suggested relationships, it is reciprocal. Thus if the relationship

   MORPHOLOGY UF MORPHEMICS

   appears in the thesaurus, the reverse must also occur

   MORPHEMICS USE MORPHOLOGY

   This relationship indicates that the term "morphology" will be used for indexing and retrieving. The term "morphemics" is treated as a synonym and will not appear as an index term on the file.

2. BROAD TERM (BT) - NARROW TERM (NT). This relationship aids in controlling the generic search problem by indicating a generic-specific relationship between terms. For example, the relationship

   MORPHOLOGY NT MORPHEMES
   NT MORPHOPHONEMICS

   implies that more precise information retrieval would be effected by searching with either "morphemes" or "morphophonemics" than with "morphology". A search using "morphology" would retrieve information pertaining to both of the narrow term concepts.

3. Related Term (RT). The related term relationship does not alleviate any of the problems discussed above. It is included in a theasaurus in order to present suggestions to one seeking information. The suggested terms are related only as concepts which may be of interest to the searcher.

The homograph problem is solved by including scope notes with terms where necessary. The term "transformation" may describe a concept both in language and in mathematics and thus would be entered in a thesaurus as

TRANSFORMATIONS (LANGUAGES)
TRANSFORMATIONS (MATHEMATICS)

Word forms were converted to a standard format by changing all plurals to singular form and all verb forms to nouns. The test items and CAI lessons were indexed using the thesaurus as the authority for applying terms. The terms were placed in the appropriate data elements on each file.

The terms which were considered as candidates for the thesaurus were gathered from the field of English grammar. A representative sample of test items and lessons were analyzed and all nouns and noun phrases were extracted to form the pool of candidate terms. The terms were then reviewed by a panel of specialists in English who arranged them according to the hierarchical relationship suggested by ANSI.

File Structure

In order to manage the large files of data from which the indexes would be generated, a computer-based indexing system was designed and programmed by the authors. This system is very powerful and flexible and, therefore, is extremely responsive to the requirements of the data being managed. The power and flexibility of the system results primarily from two design features: 1) a data element oriented file structure, and 2) table-driven programs for editing the data, generating indexes, and printing the indexes and the files.

The data element file structure was chosen to permit an open-ended file construction process. This structure allows almost unlimited addition of data elements to each record as new data becomes available. For example, a test item file could be constructed with data elements for item stem, correct alternative, and incorrect alternatives. The coordination of each item to a given course, CAI lesson, or textbook could be added to each item of the file in data elements as this type of information is identified and collected. A data element oriented file consists of records which are made up of data elements. A data element may be explained as the smallest logical and useful unit of data. Data elements are organized into records which correspond to some logical structure of the data collection. In a bibliography for example, a record would contain all the data elements pertaining to a journal article. The data elements of this record would include author, title, keyword descriptors, date of publication, and abstract. In a test item file the record would contain all data elements pertaining to each test item. In turn, this record would contain data elements such as item identification number, item stem, correct alternative, incorrect alternative, source of item, applicable academic course, etc.

For the purposes of this study a file of forty (40) computer-assisted lessons (CAI) and a file of five-hundred (500) test items in English grammar were organized into data elements. The CAI material file was organized into unique identification number data element, tutorial data element (TUT), stimulus data element (STM), and correct response data element (RSP). The tutorial data element contains passages of text which attempt to explain, instruct, and otherwise require no specific response that is intended to demonstrate learning on the part of the student. The stimulus data element contains questions, directives, or instructions which require the student to participate in a mental activity designed to promote learning. The correct response data element contains those responses which the author has indicated as correct or acceptable. In those cases where the student is expected to respond with only a letter or identifying number, the complete response has been included in the data element.

The test item file was organized into identification number data element, stimulus data elements, and correct response data elements. The stimulus includes any text such as background explanations, case histories, or sceneries as well as the question or directive to the student. The correct response includes the complete response indicated by the author of the text item.

Indexing System

The indexing system consists of three programs: 1) Edit, 2) Index, and 3) Master List. Each of these programs is table-driven, thus the user specifies at the time the program is run the requirements of the data to be edited, indexed, or printed.

The Edit program is designed to help in building the file of data to be indexed. It edits for character-set requirements, field length, data format, term frequencies, and other characteristics of the data. The edits are defined and requested by the one who is building the file and consequently may be specific to the needs of each file as it is created.

The Index program generates cross-references from among any data elements or sub-divisions of data elements as defined by the author at the time the indexing is generated. This ability to generate an almost unlimited number of cross-references at the request of the user provides the power and flexibility to accommodate the various types of data that are found in CBE materials.

The Master List produces the printing of the entire file. It is the complete reference to the content of the file. The format of the printing is designed at the time of printing.

EVALUATION

Rationale

The evaluation of the study was concerned with the effectiveness of the bibliographic indexing model, in effect, the value of the indexes in providing access to and coordination of test items and CAI materials. Thus the evaluation centered on the concepts of relevance and non-relevance as appropriate evaluation criteria. The ability of the indexes to allow users to find relevant test items and CAI lessons and to eliminate non-relevant material at the time the files were searched via the indexes was determined by computing the recall and precision ratios. The ability of a system to retrieve relevant information is known as recall and is expressed as

$$\frac{Nr}{Tr} \times 100$$

where Nr is the number of relevant items located and Tr is the total number of relevant items in the file.

The ability to retrieve only relevant information (non-relevant information is not retrieved) is known as precision, and is expressed as

$$\frac{Ns}{Ts} \times 100$$

where Ns is the total number of relevant items retrieved and Ts is the total number of items retrieved by the system.

Procedures

Printed indexes of the test item file and CAI file were generated and submitted to five English grammar instructors. The instructors were asked to request test items and CAI lessons on four topics of their choice using the indexes as the basis for selection for a total of twenty searches. The resulting test items and CAI lessons were analyzed by each instructor to determine the relevance of the selected material. Each test item and CAI lesson was ranked as relevant or non-relevant in relation to each topic.

The instructors were then asked to select test items and CAI lessons by examining the entire files without the aid of the indexes. This procedure was designed to produce the total number of relevant items and lessons in the files.

Results

The results of the relevancy judgements are presented in the tables of Figures 1-4. The relevancy judgements were summed across the twenty searches for each search. The test item file was searched using the index of item stem and the index of correct answer. The CAI file was searched using the index of stimulus and the index of correct response. The recall ratio for test item file searches using the item stem index was .83 and for correct answer index was .93. The precision ratio for item stem index searches was .82 and for correct response index was .89. The recall ratio for searches of the stimulus index of the CAI lesson file was .80, and the precision ratio was .86. The recall ratio for the correct response index was .73, the precision ratio .93.

In order to test the usefulness of the indexing model for coordinating CBE materials, the vocabularies (index heading terms) of the test item file and of the courseware file were compared. Ninety-five percent of all index headings were common to both files. This resulted in a high degree of coordination achieved via index terms.

CONCLUSIONS

The result of this study has demonstrated that bibliographic indexing techniques can be effectively applied to computer-based education materials. The standard bibliographic devices of vocabulary control through use of an hierarchical thesaurus and indexing with keyword descriptors was demonstrated to produce acceptable recall and precision ratios. These results warrant further investigation of other subject areas in order to determine any variances which may result from other vocabularies. If this further study achieves similar results, then computer-based education will have found a method for organizing, storing, and retrieving diverse materials which otherwise may not be easily managed.

TEST ITEM RELEVANCY JUDGEMENTS FOR ITEM STEM INDEXES

	Relevant Items	Non-Relevant Items	Total
Retrieved Items	250	30	280
Not Retrieved Items	50	170	220
Total	300	200	500

FIGURE 1.

272

TEST ITEM RELEVANCY JUDGEMENTS FOR ITEMS CHOSEN FROM CORRECT RESPONSE INDEX

	Relevant Items	Non-Relevant Items	Total
Retrieved Items	280	20	300
Not Retrieved Items	20	180	200
Total	300	200	500

FIGURE 2.

CAI LESSON RELEVANCY JUDGEMENTS
FOR STIMULUS STATEMENT INDEX

	Relevant Lessons	Not Relevant Lessons	Total
Retrieved Lessons	28	4	32
Not Retrieved Lessons	7	1	8
Total	35	5	40

FIGURE 3.

CAI RELEVANCY JUDGEMENTS
FOR CORRECT ANSWER INDEX

	Relevant Lessons	Not Relevant Lessons	Total
Retrieved Lessons	27	2	29
Not Retrieved Lessons	10	1	11
Total	37	3	40

FIGURE 4.

REFERENCES

Lancaster, F.W. Vocabulary Control for Information Retrieval. Washington, D.C.: Information Resources Press, 1972.

American National Standards Institute. Guidelines for Thesaurus Structure, Construction, and Use. Report number ANSIZ39.19-1974.

The Educational Technology Center at Irvine

Alfred Bork
Department of Physics
University of California
Irvine, California 92717
(714-833-6911)

Our project at Irvine has been engaged for ten years in the production of computer-aided, learning materials. The project has produced over 100 highly interactive, graphic learning modules. In addition to generating these materials we have developed an efficient and effective scheme for creating computer-based learning material. Our work has been in a timesharing environment. Support has come from the University of California and the National Science Foundation.

Important changes with regard to computer-based education are now visible. Because of the progress in developing effective material, we are at a state where we have some understanding of the medium. Coupled with this progress is the development of inexpensive personal computers. Such systems offer a likely vehicle for large-scale distribution.

Our current activities are in a stand-alone graphic environment. We are reworking the authoring capabilities and developing a new language base for these within PASCAL. We are developing new materials in this environment and will eventually transport existing materials.

Our new grant stresses interactive on-line tests, extending previous work at Irvine in mathematics, physics, and anthropology to statistics. The tests provide immediate student feedback and assistance. Several schools are cooperating in this effort; we welcome such participation. We are also developing an LSI-based central system which would keep records for on-line quizzes.

Extensive further information is available.

Development of An Evaluative Computer Simulation for Assessment of Physician Skills in the Diagnosis and Management of Patients with Myocardial Infarction.

Carl W. White, M.D., and Lisa R. Ehrlich, M.A.,
University of Iowa, Cardiovascular Center, Department of Medicine, Iowa City, Iowa, 52242.

Accurate evaluation of physicians' clinical competence by assessment of patient management skills has long been a major objective of continuing medical education. Computer assisted patient management problems appear to be an optimal method for assessing many such skills. Frequently, however, computer simulations in medical education are designed for instructional rather than evaluative purposes.

We chose to develop an evaluative simulation designed to assess both diagnostic and therapeutic skills of primary care physicians on an important topic--acute myocardial infarction. The design of the simulation allows for assessment of diagnostic skills in sections of medical history, physical examination and laboratory tests. Management skills are evaluated in uncomplicated patients, as well as in sections dealing with common complications: ventricular arrhythmias, bradycardia, hypertension, cardiogenic shock, and cardiac arrest and resuscitation.

The simulation was developed by a team consisting of a content expert (cardiologist), medical liason (physician's assistant or senior medical student), and instructional designer-programmer. Programming has been done in Basic on a Hewlett Packard 2000 access system and will be transferred for use on a Magnavox Orion microcomputer.

An extensive validation process has been designed. Medical students, house officers, primary care physicians, and cardiologists will be used to develop a hierarchical ranking system. Scoring will allow between and within group comparisons. Additional validation methods will include correlation with subject specific cognitive testing, and clinical performance ratings. Final validation will include correlation with the practice performance of physicians on this topic assessed via medical audit.

If high correlations with physician's practice performance can be obtained, the usefulness of evaluative computer simulations in medical education will be greatly enhanced.

Providing Detailed, Informative Feedback to Very Large Classes

C. Michael Levy, Ph.D.
Department of Psychology
University of Florida
Gainesville, Fl 32611

Generally the conditions for academic learning include (a) a motivated individual with the requisite intelligence and background skills, (b) a learnable task, and (c) feedback given to the individual regarding his/her performance on the task. Further, the rate of learning is optimized if the feedback is given rather quickly after any evaluation is made, and is specific and directed. Nonspecific feedback (e.g., "Your grade is B-") tells the individual if the performance is satisfactory, but if the performance is less than adequate, it conveys nothing about what should be done. In order to facilitate learning -- as distinct from performance --the person needs explicit, detailed information which elaborates the discrepancy between the actual performance and the objective. In the educational arena the student is usually given detailed feedback whenever the size of classes permits the instructor appropriate time and opportunities to provide the necessary information, conditions which seldom prevail when the class size approaches or exceeds 100.

In order to meet the needs of students in these large classes a course management system has been devised which, among other things, informs the student not merely that he scored 81% or earned a "B-" or missed items 2, 8, 17, and 22, but gives a detailed analysis of problems and errors, including explicit directions required for obtaining solutions.

The system has been evaluated using a within-subjects design where the in-class examinations served as pretests for a cumulative final examination. The pretest items were viewed only during the scheduled examination period, after which individualized, descriptive and prescriptive feedback was generated and delivered. The criterion measures included both items identical and questions highly related conceptually to those items shown earlier, plus control items focusing on subject matter taught but not previously tested. The results of this evaluation, the reactions of the students to the enhanced feedback, and the costs (labor, hardware) of implementing and maintaining the system will be discussed.

COUNSAID: A COMPUTER AIDED ACHIEVEMENT ANALYSIS SYSTEM

By: Jeffrey M. Baker
School of Medicine
MEDPREP
Southern Illinois University
Carbondale, Illinois 62901

The Outreach tutorial project is a special program of the Medical Education Preparatory Program (MEDPREP), School of Medicine, Southern Illinois University at Carbondale, Illinois. The project was initially funded in 1974 for two years through the auspices of the Special Health Careers Opportunity Grant Program (SHCOG). The project was refunded in June 1976 with a recommended three-year funding period by the Health Careers Opportunity Program (HCOP).

The project has several goals: 1) to provide academic tutoring and counseling to freshman and sophomore minority and disadvantaged students; 2) to provide financial assistance to MEDPREP students (tutors) who are pursuing a medical or dental career; 3) to stimulate interest and motivation among minority and disadvantaged students. "Disadvantaged" includes persons who are female, from low income/rural backgrounds, disabled, or persons who are veterans with medical training.

Outreach has had 229 students participate in the project since the Fall of 1974. The program encompasses a unique program in that it offers classes for credit, field trips and seminars, close association with the Career Placement Center, the Counseling Center, and the Medical School.

The program averages 50 participating students per semester, 12 MEDPREP assistant instructors (tutors), 2 graduate assistants and the coordinator. The project is housed on campus with the MEDPREP/Outreach program occupying their own building.

The Outreach program is designed to assist minority and disadvantaged students with their preprofessional courses. To do this twelve tutors are employed to assist the students in the program. Counseling in the area of skill development and career alternatives is provided by the coordinator of the project. To assist the coordinator an optical scanning form, using an OPSCAN 100DM, was developed to provide the tutors with a method of reporting the progress of the student. This data is scanned once a week and a report is produced using an IBM 370/158 computer. This report is a developmental profile of each student. The profile contains information which enables the coordinator easy access to any of 50 students' present standing according to grades, number of hours in tutoring, and number of hours missed tutoring. The profile is a valuable tool for the coordinator that provides him an early warning system for a decline in the students' performance as well as showing the strengths.

DEMO-GRAPHICS: Teaching Population Dynamics in a Multidisciplinary Framework

By Professor Paul Handler
University of Illinois
Urbana, Illinois 61801

An understanding of population, how it grows, how it changes in composition and how it affects the social and economic facets of everyday life is important to almost every social science discipline. In spite of its importance, the impact of population size and composition has been neglected because there has been no easy way to communicate the basic concepts to demography within the time constraints of present day curriculum. In addition, instructors in these other disciplines do not have the time to become familiar enough with the concepts so that they could easily integrate them into existing curricula. Therefore, a way had to be found which would give to both the instructor and the student the ideas and concepts associated with population in the shortest possible time. The program also had to have the capability of capturing the attention of decision makers in social policy within ten or fifteen minutes.

The solution to this problem lay in the field of interactive computer-generated graphics. The first programs were developed and tested on PLATO. They were rewritten many times to reorganize the material as we observed user responses. The PLATO program was used in a seminar mode in which over 13,00 foreign nationals participated. More recently the program has been translated into BASIC so that it could be made available to a much larger audience. At the present time the program is being used in over 40 colleges and universities as well as over 10 high school systems.

Examples of computer-generated output, such as age compositions and population projections as well as interactive examples, will be shown to demonstrate the teaching technique used. In particular, it will show the age composition of a country can be used to demonstrate some of its recent history. It will also be shown how family planning has changed the shape of the age composition curve of a number of countries thereby affecting school enrollments for the next twenty years. Other examples will show the demand for doctors, nurses, apartments, cars and almost any other demographic variable. The variety of displays allow the program to be used in many different disciplines including CAI. At last report the program has been used by over 10 different departments at the University of Illinois and by as many as five departments at two or three other institutions.

A Commitment to Education: Basic Math and CAI at the Aetna Life & Casualty

by

Natalie N. Lowe
CBE Consultant
Aetna Life & Casualty
Hartford, Connecticut 06156
(203) 273-2119

In April, 1978, a pilot study was completed to determine the need for a Basic Math program for employees. The 40 volunteers for the study were acquired through advertising in the weekly home office newspaper. As participants in the study, the 40 students completed a pretest of basic math skills; a 7-9 hour CAI course delivered via PLATO covering reading and writing numbers, addition, subtraction, multiplication, and division; a posttest; and a course evaluation.

Average achievement test scores increased from 58% on the pretest to 93% on the posttest. Student response was excellent with regard to media use, content, and need for math review.

As a result of the pilot study evaluation, a subsequent study covering the same material plus fractions, decimals, and percents was undertaken in July. Because the complete Basic Math course takes 10-18 hours over a period of about five weeks to complete, the second study used a sample of 20 employees. The participants were acquired in the same manner as in the earlier study. For the complete Basic Math course average achievement test scores increased from 51% on the pretest to 80% on the posttest. Once again, student response was excellent. In addition, the second study included administration of the Aiken Math Attitude Scale before the pretest and after the posttest. Significant improvement in math attitude was exhibited from pretest to posttest ($\underline{p} < .02$).

It has been determined that there is a need for remedial training in basic mathematics in the business environment. In addition to improving current job-related skills, the basic math program has the potential for opening up new possibilities for employees who had previously avoided jobs which required good math skills.

An Evaluation of CAI and Traditional Methods of Improving Computational Skills in Mathematics

Preston B. Prigger
Hollister Schools

An experiment was designed to compare CAI and more traditional methods of improving computational mathematics skills and determine the benefits, if any, of using computer based teaching.

20 seventh and 20 eigth grade students were selected on the basis of deficiencies in computational mathematics as determined from SRA scores. Each grade level was then divided into an experimental and control group, consisting of 10 students each. The experimental groups in each grade level used a computer and the control groups were given programmed work sheets covering addition, subtraction, multiplication and division. The computer program covered the same skills, but allowed the student to choose the computational area and the number of problems. The experimental groups were allowed three chances on each problem and appropriate motivational messages were also displayed. The controls were allowed to correct their errors after the teacher gave them the answers. The reward structure was similar in both groups: The computer provided reinforcing messages for the experimentals, and the teacher provided the same messages for the controls.

Although both experimental and control groups improved, the experimental groups showed the most improvement (an average of $\frac{1}{2}$ a grade level on the SRA test). The experimental groups displayed a marked increase in motivation which can account for part of the increase. The experimentals also showed a desire to "beat the computer". The controls were somewhat envious of the experimentals, but generally displayed a healthy competitive behaviour.

These results seem to indicate that a CAI system can improve computational mathematics skills beyond more traditional methods. It should be noted, however, that human interaction is a vital part of any CAI program.

THE SALES CALL SIMULATOR:
A SYSTEM FOR RAPID PRODUCTION OF SPECIALIZED TRAINING COURSEWARE

Silas S. Warner
Commercial Credit Management Corporation

This paper is to be presented as a project report to the PLATO Users Group or the Industrial Training Users Group at the 1979 convention. It deals with a system for the rapid production of courseware used by Commerical Credit Company to train sales personnel on the PLATO system. The system produces a series of PLATO lessons, each of which simulates a typical sales call situation. This system is presently being applied throughout Commercial Credit and in Control Data Corporation.

In early 1977, CCC began preparing exercises in Professional Selling Skills which made use of Commerical Credit products and real-world selling situations. This developed into a general system for gathering information from the sales force, organizing it into a simulated sales call, and preparing a lesson.

The process goes like this: preliminary information is collected from questionnaires sent to the sales force, and from promotional literature. In a special meeting, the sales force supplies the outline of a real-world sales call to the designers of the lesson. The information is recorded on tape and the call is designed during the meeting on "frame forms." The tape is then transcribed into typewritten text by a transcriber-typist, and the transcribed package is programmed into a lesson, according to directions in a manual.

The finished lesson is linked into a system of related lessons. Students have a choice of simulations, including several directly related to their product area. During a simulation, their choices are scored in an eight-dimensional array of sales skills. They have the choice of immediate or delayed feedback on this scoring. The students can also enter a diagnostic prescription generator, or notefiles, which allow rapid communication of information about the lessons, general sales pointers, and information between students and management.

This system is now in use in 26 PLATO centers designed specifically for sales training. Other Control Data learning centers are delivering sales simulators to outside companies. Seven of these centers now have the capability to create new simulations. By the end of 1978, at least 30 simulation lessons should be prepared for delivery to Commercial Credit and its subsidiaries.

Project Report: The PSO TUTOR Training Program

Celia R. Davis
PLATO Services Organization
University of Illinois

As the availability of PLATO systems expands, so does the need for training members of university faculties and representatives of goverment and industry in the mechanics and techniques of authoring on PLATO. Since 1973 the Author Training Group of the PLATO Services Organization at CERL has given TUTOR training workshops to many groups and to hundreds of individual student and faculty members. This talk will describe the goals, procedures, and materials used by the PSO Author Training Group. It is our hope that group directors, site directors, and others responsible for training PLATO authors will find here ideas which will benefit them in their own training programs.

We assume no knowledge of computers or computer programming on the part of trainees. Although our materials were designed for college students or college graduates, they have been used successfully by high school students and by college educated people in government and industry.

The materials we use are AIDS and the many on-line HELP sections, PLATO lessons which introduce the keyboard and illustrate various lesson styles, and a TUTOR training manual supplemented by lecture-discussions.

Our training program has five parts: orientation, the TUTOR editor, the TUTOR language, communications features, and PLATO resources. In addition, we cover any area of special interest to the particular trainee. Orientation includes learning to sign on and off, an introduction to the keyboard, and practice using PLATO in student mode. We begin teaching editing through personal notes. This allows learning some editing directives in a context uncomplicated by TUTOR code and introduces the communications aspects of PLATO. The TUTOR language is taught with the book Introduction to TUTOR and lesson "introtutor". Training in the language takes most of the trainee's time during the program. After it is begun, the remaining portions of the program are carried out while the trainee is reading in Introduction to TUTOR and doing the programming exercises. Trainees read the section of AIDS entitled "How to Use AIDS" and then refer to AIDS as needed during their work on TUTOR exercises. The AIDS section "Author Resources" is introduced and trainees read sections relevant to their needs. Notes, TERM-talk and TERM-consult are taught and used extensively by the trainees. Training is completed with whatever personal requirements the trainee has, e.g. site and account director options, data storage and retreival, curriculum and student routing, etc.

COMPUTER ASSISTED DENTAL SIMULATIONS

Plato Project
University of Nebraska College of Dentistry

Laurette F. Lipson, M.N.Ed.

Since June of 1977, the College of Dentistry at the University of Nebraska-Lincoln has been developing dental patient simulations under a three year grant awarded by the National Library of Medicine (LM02911). The overall goal of the project is to study the effectiveness of computer simulations and supportive library materials as a resource for improving decision-making and problem-solving skills in dental practitioners. Approximately 30 simulations will be developed in several areas of dentistry. At the date of the 1979 ADCIS meeting, nineteen problems will have been completed.

Five panels, each composed of three nationally recognized dentists in dental education and practice review completed groups of simulations for content and decision making pathways. Practitioners who participate in the Project can compare their own problem solving abilities against these pathways.

A Professional Experience Profile developed for each participant serves as a basis for the simulations he or she will experience. Supportive library materials are also being evaluated. Significant use of graphics, auxiliary slides, and touch capable responses are featured throughout.

The Project can serve as a model for disseminating library materials to a wide variety of practitioners. There are strong indications that these simulations and library materials would make excellent continuing education packages with the experience profile serving as a needs assessment document for self learning. Validation by national experts should give the simulations a broad scope for regional and/or national applicability. It is expected that the project will help identify needs in instruction in dental education and in continuing education for dental practitioners.

AN AUTHOR TRAINING COURSE FOR MILITARY AUTHORS OF INSTRUCTIONAL MATERIALS

Executive Summary

Principal Investigator

Michael Szabo, Ph.D.
Associate Director
Center for Education Research
and Evaluation
246 Chambers Bldg.
University Park, PA 16802
814-865-9500

Funded by:
The Army Research Institute
Contract #MDA903-77-C-0209

Executive Summary

The purpose of this proposal is to develop, evaluate, and revise, for use by individual military instructors, an author training course (ATC) which employs the IPISD Model as both developmental methodology and content. The stages of development of the ATC will employ the systematically applied analyze, design, develop, implement, evaluate, and control phases of the IPISD Model; the ATC will train the military author to apply the steps in his own ISD activities.

There is a pressing need to develop an ATC for IPISD for single authors for a number of reasons. First, the IPISD does not provide a system of author training. Second, recent developments in instructional research, development, and theory have led us to the point where a "design" or "linking" science between instructional theoreticians and practitioners is highly feasible. And third, a course for a single author will fill a crucial need and holds potential for expansion into an author team training course.

The three key objectives are interrelated but listed separately to highlight their individual value.

Objective 1: Using the IPISD Model, analyze, design develop, implement, and evaluate an author training course (ATC) for military instructors which:

1. trains military instructors to use the IPISD Model in their assigned instructional development duties;
2. is comprised of existing instructional materials which can be adopted, revised, or augmented for increased effectiveness;
3. provides evaluative data on the effectiveness of materials used in each block of the IPISD Model; and
4. is focused on a one-man developmental effort.

Objective 2: Using the research literature on conditions for effective learning, analyze, design, implement, and evaluate a CMI system which is characterized by:

1. real-time, interactive interface between the learner and the computer; and
2. diagnostic-prescriptive logic keyed to the student's responses to the materials stored in the computer.

Objective 3: Analyze, design, develop, implement, and evaluate off-line case studies and computer simulations as training tools and as tools for evaluation of the effectiveness of ATC and/or alternative training courses and materials which:

1. cause learners to analyze, synthesize, and integrate the knowledge and skills learned in each block and apply them to typical on-the-job tasks (OTJs); and
2. closely resemble the actions, conditions, cues, and standards actually performed on-the-job (fidelity).

A key feature is the effort to minimize costs through the techniques of: 1) selecting existing materials, where possible, 2) developing a "barely adequate" first draft course, 3) using performance data to identify blocks which need elaboration or replacement, 4) providing efficient course flow which allows trainees to bypass material where they can demonstrate prior knowledge and skills, and 5) enhancing the degree of "fidelity" between ATC exercises and activities and on-the-job performance of complex instructional tasks.

To elaborate, we anticipate the availability of approximately 80% of the total course training aids. The logistics of using these and other course components will be managed by an interactive computer system (CMI) to augment the learning. Efficient course flow will be significantly enhanced by a CMI diagnostic and prescriptive learning strategy keyed to the course objectives and instructional materials. Case studies which are managed by computer but administered off-line and on-line simulations will be developed for two purposes. They will serve as evaluative instruments to measure complex learning outcomes of ISD based on the ATC. As evaluative tools, their validity relative to on-the-job instructional development tasks will be established. The second function served by case studies and simulations will be instructional: they will enable the student to enhance the acquisition of facts and concepts through embedding and integration. As instructional and evaluative tools, the case studies and simulations will fill the large gap between acquisition of knowledge about ISD and on-the-job performance.

After the first version of the course is field tested with instructors similar to military instructors, it will be revised and implemented at a field evaluation military training site. Documentation of all materials, strategies, and programs will be provided.

COMPUTER ASSISTED INSTRUCTION! FOR NURSING EDUCATION?

Patricia J. Morin, R.N., M.S.N.
Julia A. Fisco, R.N., M.S.N.

University of Nebraska College of Nursing -
Lincoln Division

In December, 1977, three faculty received a full-funded grant of $1400 from the University of Nebraska Academic Computer Services. The project provided an opportunity to evaluate selected existing computer programs available in medical-surgical nursing (i.e., the treatment of diabetes mellitus including ketoacidosis and patient education) and in psychiatric mental health nursing (i.e., psychopharmacology, clinical diagnosis of affective illness and psychotherapeutic techniques). The programs were evaluated to determine if computer assisted learning would provide opportunities which would enhance both the didactic and applied learning experiences for students of nursing.

Special educational strategies tested were 1) problem-simulation related to situations encountered in the clinical setting (i.e., hospital), 2) testing student knowledge of content previously presented to the learner, and 3) the tutorial method of learning evaluating the student's ability to assimilate information being introduced for the first time in the curriculum.

There were 4 phases of the project: 1) developing initial tool for evaluation; obtain and review material; 2) completing evaluation of materials and develop an initial plan for use, 3) discussing potential use with faculty, and 4) submitting final report of project.

The benfits of the project for the College of Nursing, the University of Nebraska, and the general public include 1) providing evaluative information to the faculty involved in curriculum development and implementation, 2) providing an impetus for computer program development, and 3) increasing student competence in the actual clinical setting, thus benefitting the general public who are consumers of health services.

Herbert Bielawa
San Francisco State University

GRADING UTILITY SYSTEM MANAGER

Grading Utility System MANager is a Basic program created for the purpose of allowing students to monitor their course progress at any time. In order to use this program a course must be based on a point system. It has two parts: GUSMAN and GUS. GUSMAN is the read - write portion serviced by the professor who enters points for exams, projects, reports or whatever. The functions in GUSMAN are:

ADD (names, projects or scores)
CHANGE (names, codes or scores)
LIST (listing of points for all projects)

An important entry to the class student list is CONTROL, which registers the hypothetical maximum for any project. GUSMAN then compares all student points totals with CONTROL, readjusting their status as each new set of points is entered. This is given in percentages (as well as in total points). At the conclusion of the term, when final points are entered, GUSMAN exhibits final points totals and percentages. However, the simple percent figure is sufficient to translate directly into letter grades.

GUS is a read - only subsidiary program which is run by students. It pulls from GUSMAN just that information which applies to the individual student. The student printout shows a list of projects and points (with entry dates), as compared to CONTROL.

With GUSMAN, the chances for uncomfortable grade "surprises" for the student are minimized. The student knows what his/her final grade is at any time during the semester by reading his/her GUS file.

PROJECT REPORT TITLE: "NSF CAUSE Contribution to Computing in Natural and Social Sciences at Muskingum"
SITE: MUSKINGUM COLLEGE; New Concord, Ohio 43762
DIRECTOR: James L. Smith (614-826-8306 or -8211)
ASSOC. DIR: Larry J. Zettel (614-826-8301 or -8211)
HARDWARE: PDP 11/45 (RSTS/E)
LANGUAGES: FORTRAN, BASIC, COBOL, Assembler, Others.

The Muskingum College NSF supported CAUSE project is a computer literacy program which started in June, 1977. Objectives of the project include (1) have all students able to demonstrate a modest level of proficiency in using statistical and/or algebraic analysis with the computer, (2) have all science students demonstrate ability to apply methods and techniques learned to a varied set of simulated 'real world' problems, and (3) have all science students and faculty utilize the computer with significantly increased frequency in classroom and laboratory work as a (a) problem solving tool, (b) teaching-learning device, and (c) communications resource.

The academic departments included in the project are: Biology, Chemistry, Economics and Business, Geology, Mathematics and Computing Science, Physics, Political Science, Psychology, and Sociology-Anthropology. Major project activities to date have included purchase of equipment, faculty preparation (two 2-4 week training sessions), implementation of the computer in courses in eight of the nine departments, and evaluation.

This report includes (1) summary of equipment on site, (2) specific uses in redesigned courses, and (3) midterm evaluation and summary information. In addition to examples presented in (2), others will be available on a handout.

Title: Description of the Physiology Self-Evaluation
Units (PHYSEU) Management and Reporting System
at The Ohio State University College of Medicine

Hardware: IBM 370/158

CAI Language: IIS (Interactive Instructional System)

Contact Person: Verl E. Dennis, Ph.D., Division of
Computing Services for Medical
Education and Research, 076 Health
Sciences Library, 376 West Tenth
Avenue, Columbus, Ohio 43210

The PHYSEU program was initiated at The Ohio State University College of Medicine in 1976 as a joint project of the Department of Physiology and the College's CAI Development Staff. The program is a major component of the overall instructional approach to the teaching of physiology and is designed to provide students with a means of self evaluation. The CAI program consists of 15 content units and 4 review units. An on-line management and reporting system for the instructor was implemented to accomplish the following goals:

A. To provide the instructor with a variety of status reports for purposes of instructional management;
B. To provide the instructor with a set of commands that can configure the program so as to adapt it to a broad range of curricular programs. For example, the instructor may determine if scores on the PHYSEU units are to be recorded; determine what units are to be presented and the sequence of their presentation; determine if the student's progress is to be halted if a criterion score is not met, etc.;
C. To provide a set of commands that permits the instructor to create and maintain student records; for example, change a score.

The PHYSEU commands permit the instructor to fashion the PHYSEU program so it can serve a number of instructional needs without resorting to programming the unique requirements for each different instructor or curriculum. Therefore, a variety of users with different instructional needs may easily construct their own PHYSEU program.

COMPUTERIZED PATIENT MANAGEMENT SIMULATIONS IN PEDIATRIC CERTIFICATION

Ernest N. Skakun

Recently a variety of simulation techniques have been introduced to measure clinical skills of medical trainees. A commonly used simulation is that of patient management problems. One modification made to patient management problems is to adapt them for presentation via computer systems.

Computerized patient management simulations have been used on an annual basis since 1974 as a complementary testing procedure for the certification of pediatricians by the Royal College of Physicians and Surgeons of Canada. Each examination consists of four problems, each of which takes the candidate through a history, physical examination, laboratory requests, and a treatment and management section. The examination is available in the English and French languages and is usually administered in a number of centers across Canada.

The work with the computerized examination suggests that it is technically possible to conduct the examination in a variety of centers across Canada. A comparison of candidate performance on the computerized simulations with performance on other examinations such as multiple choice and orals reveals a low positive correlation. If it can be assumed that the measures are valid and reliable then this low correlation could be interpreted as an indication that the simulations measure candidate skills not assessed by the other examinations. Candidates when questioned about the computerized simulations expressed a favorable attitude towards this form of examination.

Currently, validation studies of the computerized simulations are being conducted.

PLATO-ASSISTED TACTILE SPEECH: DATA FROM FIELD EVALUATIONS

Frank A. Saunders
Theresa A. Easley
William A. Hill

Smith-Kettlewell Institute
San Francisco, California

1. Intended Audience: SIGED

2. Purpose: A PLATO-based curriculum is teaching profoundly deaf children to understand a vocabulary of spoken words and phrases, in conjunction with a new sensory aid. The aid analyzes and displays sounds as a tactile pattern, on a belt of stimulators worn around the abdomen. This paper presents the results from an ongoing field evaluation of the sensory aid and the PLATO curriculum, as experienced by six profoundly deaf children in a public school classroom.

3. Method: The functioning of the sensory aid, and the goals of the curriculum, were described in a topic presentation at the 1978 ADCIS meeting. Briefly, the aid analyzes sounds according to frequency; the 32 tactile stimulators, each responding to a specific pitch, create a dynamic touch pattern corresponding to the acoustic characteristics of speech. PLATO, in conjunction with a random access audio unit, presents words and phrases, associated with graphic illustrations and text. The curriculum presents spoken material for vocabulary drill and testing, in a graded sequence according to phonemic difficulty, response set size, and syntactic complexity.

4. Results: Six profoundly deaf children, ranging in age from 5 to 8, have received training in tactile speech perception since March, 1978. We are obtaining reliable data regarding acceptability of the aid, rate of vocabulary acquisition, phonemic discriminability, and the effects of background noise. These data will be presented and discussed in the project report.

Design and Implementation of a Large Scale CMI Project on a Microcomputer System

Contack Person: Harry W. Alderman, II
Coordinator of Applied Mathematics
Community College of Denver
Golden, Colorado 80401
303-988-6160 Ext 223

Hardware: Z-80 (Zilog) based microcomputer system with 50K of 250 ns semiconductor memory. Two 8" floppy disk drives each capable of 256K bytes storage. Four digital tape drives. One Decision OMR 6500 optical mark reader. General Electric Terminet 30 provides hard copy output for student and instructor use. The above is configured as a totally intercept driven, multitasking multiterminal, real time system. The software was developed utilizing a Z-80 machine code assembler.

Student Handling Capability based upon two disk drives:

1150 total students
24 different classes which require diagnostic testing and complete diagnostic information on each student

An unlimited number of classes may be run on the system if diagnostic pre-testing is not required. Note that total number of students may not exceed 1150 for two disk drives.

256 total test keys may be stored for automatic grading utilizing optically scanned input. 24 diagnostic test keys may be stored for automatic grading utilizing optically scanned input. These keys provide complete diagnostic information to both student and instructor.

Typical Student Record Access time: 3 sec.

All of the above parameters may be increased by adding additional tape drives.

Hardware and Software for this system have been designed to allow for the incorporation of two additional central processors at a later date. This parallel processing configuration will allow for vastly increased system flexibility.

Some items contained in Student Record are:
Complete diagnostic results, instructional block results, student instructional time, test dates, entry and exit dates, social security number, name, address, phone number, course number and section.

Information contained for administrative and general instructional purposes include:
Laboratory utilization times to be used to determine staffing trends, automatic letter generation for non attending students, automatic grading and class roster generation. A report generation package is contained to allow flexible formatting of required administrative reports. A statistics package is available for loading during off hours. This package allows for test item analysis, student demographic profiles and test reliability measures among other things.

All systems and student files require security codes for entry. All student files are fully backed up immediately to provide insurance against system malfunctions. A cyclic redundancy check method of recording is used to provide a virtually zero error rate in recording student files.

Stewart A. Denenberg
SUNY Plattsburgh

An Alternative Curriculum for Computer Literacy Development

An Alternative Curriculum for Computer Literacy Development (ACCOLADE) is a system that allows people to use computers and people to educate themselves about computers. The goals of ACCOLADE include: allowing the Learners to take responsibility for effecting their own education, encouraging a Teacher-Learner partnership, identifying and providing resources for the acquisition of computer literacy knowledge in the areas of applications, history, social issues, hardware, and software; illuminating the structure of this knowledge space and, by doing so, shaping an informed attitude about what computers can and cannot do, and teaching the skills and behaviors of logical thinking, problem solving, learning, confidence and autonomy through the medium of computer programming.

The ACCOLADE system comprises Learners, Teachers, and a Computer. The Computer consists of three components: a Communications File, a Map, and a Computer Managed Instruction (CMI) system. The Map, inturn, is broken down into three parts: a Yellow Pages, a Dictionary and a Keyword Index.

The Learner acquires computer literacy knowledge and programming skills by appropriate interaction with the Teachers and the Computer. The Teacher assists and guides the Learner and manages the Computer component of the system. The Computer affords a CMI system to allow self-paced learning and testing for the Learner, a Communications File to facilitate Learner/Teacher and Learner/Learner Communications, and a Map which allows the Learner to take responsibility for acquiring computer literacy knowledge. The MAP shows the relations between the computer literacy topics using a linked semantic information network (called a Dictionary), Keyword Index, and Yellow Pages of resources for computer literacy knowledge.

ACCOLADE was implemented in a pilot study using ten Learners, two Teachers and the Control Data Corporation PLATO IV Computer System. The pilot study was evaluated as reasonably successful in terms of the goals of ACCOLADE and indicated that the computerized knowledge part of the system could, in theory, be extended to domains of content knowledge other than computer literacy.

DESCRIPTION OF A COMPUTER BASED INSTRUCTIONAL TV SYSTEM: DAVID

James R. vonFeldt, ITV/C Specialist, National Technical Institute for the Deaf, also president of VONTECH Inc.

The DAVID system (Digital And Video Interactive Devise) was designed to maximize the effectiveness of video based instruction. The system consists of microcomputer, video tape replay system, color television (CRT), keyboard and mini disk.

DAVID instruction is designed to control color video tape (with audio), allowing the instructor to present segments of a stimuli as well as reinforcement, using the video media. Student interaction, answer analysis, branching and other accounting are done by the computer as would be expected in a CAI system. Single word, sentence, or selected words in a sentence can be matched thus branched to a reinforcement and or the next instructional segment. Text reinforcement can also be used.

The first DAVID system was implemented at the National Technical Institute for the Deaf, Rochester, New York, in November 1977. This system was designed and is dedicated to research thus requiring special features such as ability to communicate with other computer systems, 10 mb disks and a printer. Research in speech reading skills is underway at this time. In contrast to this system the production DAVID system is a stand-alone portable system.

Over 500 people have experienced DAVID instruction since implementation in November of 1977. The system attracted particular attention at the HEW Birthday celebration particularly because of the children's interest in the language TV tape segments (adapted to DAVID instruction) developed by Dr Margret Withrow and Dr. Charles Csuri. In addition, three strategies were demonstrated in the NTID demonstration lesson: 1. drill and practice, 2. gaming, and 3. dialogue (social simulation). DAVID instruction is particularly well displayed in the social interaction segment where you are being interviewed for a job under very real conditions.

We are calling these programs DAVID instruction as they depart significantly from linear instructional television or the limits of CAI (even with color line graphics). In a way the name was chosen for its symbolic meaning: small and very powerful. Even so, NTID has successfully adapted existing TV for DAVID instruction.

Production systems are available for approximately $15,000 per system from VONTECH Inc., Wyoming, New York, 14591.

PROJECT REPORT

The Qualitative Optimization of Drill and Practice Programs for the Deaf

Greg P. Kearsley
Courseware, Inc.

In the fall of 1977, drill and practice programs in arithmetic were provided to elementary students at the Alberta School for the Deaf via a terminal connected to the University of Alberta. One of the major goals of this project was to study the qualitative optimization of the programs, i.e., the process of tuning the response analysis and logic to catch increasingly more subtle errors, misunderstandings and misconceptions. This amounted to improving the program's "understanding" of what the student was doing wrong and providing immediate remedial instruction rather than trying to have the program match the students difficulty level based upon past performance scores (qualitative optimization).

Qualitative optimization can be conducted at two levels: a surface or syntactic level which involves making the programs smarter with respect to content-free errors, such as the handling of inappropriate responses, and a semantic level which involves creating diagnostic routines and remedial sequences to deal with student misconceptions in arithmetic, such as problems in carrying digits. Surface level optimization was important with this group of day students, since they often had difficulty following instructions or requesting help, and the ability of the program to detect inappropriate responses (e.g., characters instead of numbers, missing symbols, etc.) was needed. While insufficient data was collected to discover if deaf students require different types of error diagnosis rules in arithmetic practice, it appears that they make some of the same mistakes made by hearing students.

Qualitative optimization of CBI programs leads, not only to more effective instructional materials, but also to a better understanding of the errors which students make in learning a subject. For this reason, further research in this area is encouraged.

COMPUTERIZING THE INSTITUTE OF AVIATION AT THE UNIVERSITY OF ILLINOIS

Stanley R. Trollip

The Institute of Aviation at the University of Illinois trains both pilots and maintenance personnel at the University-owned airport. Each year several hundred students are certified by the University under authority from the Federal Aviation Administration (FAA). In order to automate much of this training five major computer-based projects have recently been started.

1. All the groundschool classes associated with the various flight courses are being reorganized to run on a computer-managed system implemented on PLATO. In addition, various types of testing will be computer administered. For example, all advanced students are already required to take and pass at the 85% level a computer-based test equivalent to the FAA Private Pilot Written Examination. Failure results in the withdrawal of pilot-in-command priveleges until the test is successfully taken.

2. Various components of the maintenance curriculum are being transferred to a computer-managed system implemented on the University's Cyber system. This includes CAI components such as welding simulations.

3. Several PLATO CAI programs are currently being used in an instructional mode in the flight training curricula. These programs range from introductory lessons to complex simulations which teach instrument maneuvers.

4. A computer program is being written to handle all the scheduling of the thirty aircraft in the training fleet. The program will schedule aircraft to students in most need of flight time, taking into account such factors as weather and grounded aircraft.

5. A Zylog Z80 microcomputer system is being interfaced with a desktop flight trainer to investigate the feasibility of stand alone flight instruction. Studies are also being conducted to determine the role such microcomputer systems may play in groundschool training.

Robert Foster
Duane Stevens
San Diego City Schools

CORK

Computer Operated Record Keeping in selected Compensatory Education Schools of San Diego City Schools.

CORK is the system devised in San Diego City Schools to relieve the teacher of excessive record keeping responsibilities.

The computer record keeping process (CORK) due to the success at the one junior high school was expanded to five elementary schools in 1975-76. The system was not successful at the elementary level. Causes for failure were conjectured to be as follows:

(1) Lack of commitment to the system by at least one person at the site.
(2) Record keeping at elementary sites was primarily a classroom teacher task rather than at a central math lab.
(3) Frequent computer breakdown.
(4) Delay due to an overloaded system.

In 1977-78 the program was revised according to recommendations to avoid an overloaded system. A card reader was added so that tests could automatically be corrected while simultaneously updating the files; and the system was piloted at one elementary site with a committed teacher.

The results of these corrections have demonstrated a more workable helpful system which actually relieves some of the classroom teacher's burden.

CORK now provides:

(1) Automatic test scoring
(2) Automatic student profile updating
(3) Continuous feedback on student progress for teachers and students
(4) Up-to-date summaries of class and student achievement
(5) Additional personnel time for the individualization process through less time required for record keeping.

THE DEVELOPMENT OF A COMPETENCY-BASED EDUCATION PROGRAM IN MUSIC THEORY

Michael Arenson
University of Delaware

During the Spring Semester, 1978, the music theory division of the Department of Music at the University of Delaware was awarded a grant from the University Center for Teaching Effectiveness to develop a competency-based education (CBE) package to be used with the core harmony and ear-training courses taught during the first two years of the undergraduate music curriculum. The delivery system selected for the CBE program is the PLATO system. The PLATO Computer-Managed Instruction (PCMI) package developed by Control Data Corporation was chosen to provide the framework for the diagnostics and learning resources for the CBE program.

During the Summer Semester, 1978, the music theory division began the development of a competency-based format for the first harmony and ear-training courses. This development included the formulation of objectives and mastery levels for the modules to be taught and the production of these modules. It also included the designing of diagnostic tests and learning resources.

During the Fall Semester, 1978, the process of entering the CBE modules on the PLATO system began. Also begun at this time was a systematic revision of each of the modules. At the beginning of the Fall Semester, a pretest covering all materials in the first-semester harmony course was given to all students in that course. The same test was designed to serve as a posttest at the end of the semester and will also serve as a pretest and posttest next year when first-semester harmony students proceed through the competency-based format for the first time. A comparison of students from this year's noncompetency-based first-semester harmony course with students from next year's competency-based course will be made to examine the success of the competency-based format in teaching music theory.

COLLEGE GRADE INFORMATION COMPUTER SYSTEM

Ira L. Karp
Virginia State College
Petersburg, Virginia 23803

Laenu A. G. Karp
Mercy College
Miami Center for Bilingual Education
Miami, Florida 33130

Present classroom procedures require instructors to test students at appropriate intervals, grade and record the results, and communicate these results to students, parents, and administrators. Usually, only the midterm and final grades are thus reported because of tedious and time-wasting techniques. This procedure provides needed information, including objective testing and evaluating procedures; specific grading procedures; 24 hour (or less) turn-around time of test results; frequent, easily graded, short tests; automatic recording procedures for quizzes as well as mid-term and final grades; grading procedures easily transferable from one institution to another; separation of the testing and grading function to help insure greater objectivity; greater security in handling test materials; provision for unexpected irregularities in attendance; flexible application of these grading procedures; transmission of information to student, parent, and administrator concerning potential problems students are having; progress reports to parents, administrators, and students; detailed statistical information, especially to administrators concerning student progress; maintenance of student privacy. Appropriate COBOL programs and file structures have been written for use on an IBM 370/135 DOS system. This program presently handles: weighted average grading schemes; excused absences; the dropping of certain test grades; and sophisticated curving. In progress of solutions are problems of item analysis; individualised outputs to students showing progress being made; direct access to test-data bank stored on disk; tape back-up storage of test grades; and techniques for handling improperly punched input cards.

Computer-Based Educational Technology Program
at the
Lister Hill National Center
for Biomedical Communications

Charles M. Goldstein and William H. Ford, Jr.

National Library of Medicine
Bethesda, MD 20014

INTRODUCTION

The Lister Hill National Center for Biomedical Communications (LHNCBC) of the National Library of Medicine (NLM) has supported a program in Computer Based Educational Technology (CBET) to develop a wide range of economical computer-based systems and extensions to existing systems for purposes of improving the effective delivery of computer-based educational materials (CBEM). These systems employ recent advances in mini/microcomputer auxiliary computational hardware and storage technologies. The purpose of this paper is to describe the major LHNCBC projects and their present status. In order to present the LHNCBC developmental efforts in proper perspective a model Learning Resource Center is depicted in Figure 1. The basic elements -- Central Catalog Control and Delivery System, Communications, Online Access, and Stand-alone Delivery Systems -- represent LHNCBC CBET projects, in a variety of configurations and their interrelationships.

Conceptually the model in Figure 1 depicts a Learning Resource Center (LRC) on a medical or health sciences campus. The Central Catalog Control and Delivery System (or Central System) may be, typically, a minicomputer supporting the authoring, delivery, and instructional management of CBEM, as well as access to other network resources, on or off campus. It may also service subsidiary LRCs in a multi-campus institution. The potential for a centralized catalog and aggregated student statistics have obvious advantages.

It is to be expected that a large variety of computer terminals will be utilized for online access. While not all such terminals will support computer-generated graphics, there will be a common need for auxilliary devices for ancillary graphics such as slides, videodiscs, etc. The Stand-Alone Delivery Systems will allow of both online access to courseware as well as the "downloading" of courseware for stand-alone operation. This latter represents the bottom rungs of a hierarchical delivery system. The upper rungs may be regional, national, or network-shared catalogs of courseware accessed by the LRC-Central System.

An actual LRC might include only a subset of the capabilities in Figure 1 initially. The description of LHNCBC projects relating to the model LRC is organized into six sections. They are:

I Communications

II Central Catalog Control and Delivery System

III Auxiliary Devices

  A. Slide Projectors
  B. Videodisc

IV Delivery Systems

  A. Network
  B. Distributed

V Access to Library and Information Systems

  A. Information Transfer System
  B. Integrated Library System

VI Conclusions and Future Direction

An important aspect of the LRC of the future will be both the modularity and the level of integration that can be achieved between the various subsystems. A major goal of the program at LHNCBC is the sharing of resources through system integration. Subsystems that presently require specialized components are being merged with other subsystems to create a general purpose capability. Several examples of successful systems integration will be presented.

I. COMMUNICATIONS

A prerequisite to the design and implementation of a fully integrated LRC is the ability to support a wide range of communications protocols and provide for data channel routing (switching) between resources. A key to resource sharing is the ability for systems to communicate with others in the same environment. The LHNCBC has been involved in the design and development of two subsystems which directly address this need.

o Automated Data Switch (ADS)

A problem commonly encountered in an LRC is the limitations placed on the individual terminals. Terminals are often "hardwired" to a host computer or network. The ADS provides a means of interconnecting any group of devices for a given session. As an example, a display terminal normally connected to a local computer could request a line to an external network; the ADS would locate an available line and dial the phone. In addition, the user might desire a hardcopy of the session. The ADS could connect an "available" printer to monitor the session. On completion, the ADS would return the outgoing line and printer to the "resource pool." While a prototype ADS was previously developed, LHNCBC is presently working with industry to develop a model installation for our new facility, scheduled for March 1980.

- Intelligent Input/Output Buffer (IIOB)

Although the ADS solves the problem of interconnecting devices, it does not provide for speed, code, or other channel dependent format conversions. The IIOB was designed specifically to provide a low-cost (ca. $600 per unit) resolution to these and other communications interface problems. As an example, the IIOB could be placed between a low speed asynchronous terminal and a high speed synchronous network interface, thereby, removing the requirement for specialized terminals. The LHNCBC has several major developments underway which include an IIOB. At present these developments are primarily directed at distributed library systems. A prototype application IIOB will be demonstrated in the fall 1979 as part of a major library automation project. It should be pointed out that the Intelligent Device Controller, discussed in a subsequent section, is, in fact, a special version of the IIOB.

These two subsystems offer a level of integration and resource sharing not yet achieved in a LRC. Terminals, printers, and other commonly used devices can be dynamically configured. External communications channels can be shared. If multiple computers are used within the LRC environment, a lower number of connections will exist through effective communications resource management. An effective communications subsystem provides an economic incentive for systems integration. The LHNCBC program is directed at demonstrating these and other alternatives to dedicated subsystems.

II. CENTRAL CATALOG CONTROL AND DELIVERY SYSTEM

Having established an environment for resource sharing, the next step in the development of an integrated LRC is a mechanism to manage or control the delivery of courseware. An important concept in our model LRC is the potential separation of courseware management and authoring from actual execution. As microcomputers become more powerful, the need for a large centralized computer, like PLATO (1), diminishes. As reported in Electronics, Annual Technology Update (2), Very Large Scale Integration will make significantly more powerful microcomputers and minicomputers available by the mid 1980s (16 bit microcomputers in 1980 and 32 bit by the mid 1980s). Based on these and other earlier predictions for hardware/software advancements and an appraisal of the problems that would be encountered in a LRC which utilized a wide range of CBEM, led to the following developments:

- Minicomputer-based LHC-PILOT (3, 4)

  Historically, Computer-Assisted Instructional (CAI) Systems have been limited to a single language. This has been especially true in the minicomputer systems. Although LHC-PILOT is the "host language," courseware written in other languages can be managed and distributed by the LHC-PILOT System. From its inception LHC-PILOT has had five goals:

  1. Support a complete online interactive courseware development capability including collection of student statistics.
  2. Provide the necessary extensions to CORE-73 PILOT (5) for the development of courseware which includes simple file input/output, graphics, and improved answer judging capabilities.
  3. Provide for the use of distributed program execution as an integral system component.

4. Allow for the management and delivery of non-PILOT courseware on stand-alone systems.
5. Execute on all versions of the host minicomputer system, from microcomputer to a very large minicomputer.

Specific examples of these and other system features are discussed in the subsequent paragraphs on the LHC-PILOT System. The present Release (3, 4), Version 1.1, does not include all these features. Release 2.1 does include the major features discussed and will be available in the first quarter 1979 for test. The complexity of the LHC-PILOT System is such that a detailed description cannot be included in this program update. The features discussed provide the user an overview of the system's capabilities. Emphasis has been placed on those features which directly relate to the development objectives described previously. The presentation of the LHC-PILOT System is in three parts:

1. System Administrator
2. Authoring Subsystem
3. Course Management and Delivery Subsystem

System Administrator

The System Administrator (SA) is a special class of user in the LHC-PILOT System. The SA manages the LHC-PILOT operating environment, has access to, and controls information files regarding system utilization, accounting data, user password, and directory/space allocations.

Authoring Subsystem

The Authoring Subsystem (AS) is complete in that no external system resources are necessary for an author to create, maintain, and deliver CBEM. Although not as sophisticated as the larger systems (PLATO, Coursewriter), it has many special features not normally found in a CAI system, mini or maxi computer version. Considerable design and development has been directed at the needs of an author of LHC-PILOT courseware. The following summarizes a few of the major features in the AS:

1. Interactive debug including direct command execution
2. Self contained course editor with a full set of commands (Search, Change, Up, Down, Top, List, Print, Size, etc.)
3. Realtime syntax verfier. Each line entered is verified on input and messages are displayed for syntax errors detected.
4. A series of special commands are provided to aid in the location and storage of new or modified courses.
5. An author may have multiple directories, a series for development and a series for student use.

The ability to interactively test commands, especially answer judging and graphics, provides an author with realtime feedback on his or her ideas. New concepts can be tested without anything other than an online terminal at the author's desk; results are immediate.

These and other tools are necessary to remove the burden presently encountered in many systems by new authors. In addition, a highly interactive and responsive system encourages the testing of new ideas. Simply stated, it is much more fun for the author than most CAI systems.

Course Management and Delivery Subsystem

The Course Management and Delivery Subsystem (CMDS) is unique in its approach. The classic systems, COURSEWRITER (7) and PLATO (1), have historically supported course execution on the host.

311

PLATO V (8) offers a limited capability to remove simple processing tasks from the host computer, but requires the user to remain online. The LHC-PILOT CMDS incorporates two major features necessary to achieve the required level of systems integration in our model LRC. They are:

1. During LHC-PILOT course development, the author may choose one of the following execution options:

    a. Direct execution online to the LHC-PILOT System
    b. Distributed execution in an ATS (Advanced Terminal System, previously called Plasma Terminal System)*
    c. Downloading of courses for stand-alone execution in an ATS.*

2. The second and most important feature of the CMDS is its ability to manage and deliver (download) courseware to non-PILOT delivery systems. Programs written in MUMPS (9, 10), BASIC, and other interpretive systems are ideal candidates for a centralized courseware distribution center. Future developments should not be limited to interpretive languages. By maintaining an up-to-date directory of available courses, independent of language or host computer, the LHC-PILOT System provides a single source for information on CBEM in the LRC. Further, providing a centralized distribution center for courseware received from outside the LRC insures up-to-date releases for all courseware users. A further advantage of this distributed computing, centralized courseware management concept is the collection of statistics related to a given course.

* Refer to section on Delivery Systems in this paper for additional information on the ATS.

Since course changes are downloaded, it is possible to upload the daily/weekly student responses for analysis. LHNCBC plans to investigate the data management problems associated with these and other responses from distributed systems.

The LHC-PILOT System is only one of many alternatives available for centralized management, decentralized delivery of CBEM. The LHC-PILOT System does offer a relatively complete test bed to evaluate these and other new concepts in a single system configuration.

III. AUXILIARY DEVICES

An important function of an LRC is to provide access to a wide range of materials, often produced on different media (slide/tape, videocassette, CAI etc.). In many cases if access to the materials were integrated and made available in a single learning carrel, more effective utilization would be achieved. Our model LRC has the capability to deliver 35mm, video, and audio sequences under computer control. The primary mechanism is an Intelligent Device Controller (11). The Intelligent Device Controller (IDC) can take on a variety of tasks. Figure 2 shows a typical configuration of a terminal, IDC, two auxilliary devices, and a host computer. Two of the devices presently supported by the IDC are:

1. Random Access Slide Projector - There are several major CAI programs which utilize 35mm slides. The student is required to manually position the display unit for each reference made by the computer. Our LRC has all slide projectors interfaced to an IDC for use by local or remote computers

as illustrated in Figure 2. The IDC is transparent to the terminal and host communications under normal operation. The IDC only acts on data preceded by a user defined code.

2. Videodisc - A new technology that holds promise for future CAI and information systems is the Videodisc (12). A major part of the IDC design has centered around the unique capabilities of the videodisc. The major features of videodisc technology are:

   a. Randomly accessible audio and/or video sequences
   b. Storage and random access to tens of thousands of color stills on a single 12 inch disc
   c. The potential for extremely high density digital storage (13) that can be mass produced.
   d. Mass production of videodiscs at very low cost (less than $10 in quantities of a thousand)

The concept of a single "black box" capable of controlling all auxilliary devices available in the LRC provides an economic incentive to purchase only devices which can be so controlled. The design of the IDC is such that new devices can be interfaced without great difficulty. In addition, the software is designed to allow field updates with relative ease. Any vendor supporting the IDC will, no doubt, have a "catalog of devices" available.

The IDC development is to demonstrate the feasibility of a single "black box" controlling a wide range of equipment types without major modification. The prototype IDC is presently under test at LHNCBC. The present cost in single units is approximately $600. If mass produced and marketed, it is expected that the price would decrease substantially.

IV. DELIVERY SYSTEMS

There are three types of Delivery Systems (DS) available in our model LRC:

1. National Networks (PLATO, Health Education Network, etc.)
2. Local Multiuser Systems (LHC-PILOT, MUMPS, etc.)
3. Stand-Alone Systems (LHC-PILOT, MUMPS, T-TUTOR(?), etc.)

A major goal of the LHNCBC program is to remove the requirement for dedicated hardware to access the major CAI systems. The Advanced Terminal System (6) is a prototype system designed to support the three primary types of courseware delivery listed above. As discussed in the Lister Hill Center Plasma Terminal System and Implications for Medical Networks (6) and Advanced Terminal System (ATS) Users Guide (14), the ATS is directed at removing the requirement for specialized hardware for each type of DS utilized. The ATS is, in fact, a stand-alone DS. In summary, the ATS includes the following features:

* A stand-alone set of graphics primitives that include the primary PLATO graphics commands as a functional subset
* PLATO V compatibility
* A Graphics Primitive Language (GP/L) that is host language independent
* An extended PILOT interpreter which includes GP/L syntax processing
* A remote computer interface which includes GP/L syntax processing

* A Diskette Operating System (DOS) capable of meeting the CBEM delivery requirements of stand-alone and distributed CAI systems
* A GP/L processor interface to support the implementation of other microprocessor-based languages (e.g., MUMPS)
* A Download/Upload subsystem for the transfer of programs or graphics data to or from a remote computer system.
* A self-contained set of development tools for the creation, updating and viewing of locally designed CAI programs and graphics data
* Two classes of locally stored graphics images, (1) DOS based graphics frames for archival storage, and (2) frame buffer memory for interactive and repetitive display of DOS images
* Locally applied transformations to frame buffer
* Two optional hardware arithmetic processors for simulation and advanced graphics
* A modular hardware/software design, which facilitates field upgrades and future system expansion; e.g., system memory, DOS memory, and communications modules can be added as required to the existing system.

At present there are two prototype ATS systems under test. A major goal this fiscal year is to demonstrate the power of the ATS both in network and stand-alone environments. Considerable efforts are being placed against performance testing and benchmarks. The tests and benchmarks are designed to evaluate the ATS's capability to function as a small stand-alone simulation system. Another area of development is a CRT replacement for the plasma panel. A prototype will be available for test, interfaced to an ATS, in early 1979. A final report will be prepared in 1979 on the ATS development. Planned extensions include replacement of the floppy disk with recently available disk system containing 14 to 70 megabytes of storage for less than $5,000.

V. ACCESS TO LIBRARY AND INFORMATION SYSTEMS

A unique feature of this LRC is the ability to allow students (users) access to local (campus) and remote (off-campus, national networks, etc.) library and information systems from the same computer terminal used for CAI. This feature will support access to reference information and data during and/or as part of the educational process. While the basic, technological means to accomplish this are inherent is the communications capabilities of the Central System as depicted in Figure 1, and described elsewhere (see COMMUNICATIONS), two other developmental efforts at LHNCBC serves to emphasize this potential. They are the Information Transfer System (15) and the Integrated Library System (16).

- Information Transfer System (ITS)

  At present the primary objective of the ITS is the delivery of up-to-date information on major diseases such as Hepatitis. It is, in reality, a functional model of an online encyclopedia with full-text information and computer generated graphics. The addition of a videodisc and Intelligent Device Controller (q.v.) will enable the addition of color "plates" (pictures) to this model encyclopedia, as well as randomly accessible video sequences in full color with sound. The latter is a feature not shared by "hard copy" encyclopedias. The system overview in A Full Text Data Base With Graphics (17) describes the basic Delivery System capabilities.

This type of "Information Bank or Online Encyclopedia" could play an important role in CAI. Courses could "refer" the student to online data concerning current test results and/or research data. The total system is under test at the LHNCBC. There are three microprocessor-based, stand-alone prototype Delivery Systems in the field for evaluation. In addition, remote terminals access the same data base on a minicomputer at NLM.

- Integrated Library System (ILS)

  The ILS is a major development to demonstrate that a fully integrated library system can be designed and implemented on small-to-medium scale minicomputer systems. One of the more exciting features of the ILS will be the User-Cordial Interface (18). Library users have historically used the manual card catalog to locate information. Little or no automation is available today to aid the Medical College library user in qualifying and then locating a specific item. The significance of LRC access to the ILS can be readily seen when a student taking a course is "referred" to a specific article, chapter, or complete book in the library. Online and interactively, from the LRC, the student will be able to search the library for a specific item. Once qualified, the student may have it put on Hold for pickup, mailed to him/her, or placed on Reserve (waiting list). In addition, when a student encounters a course that covers a specific subject area, but lacks in background or examples, he/she will be able to search an ILS data base for "references" on the material being presented. Once qualified, the item(s) can be processed as previously described. The concept of online access to a local ILS, which has access to all campus library data bases, will present a new dimension in educational services.

  When completed, the ILS will also allow access to network resources in a manner as previously described for the LRC. In our model library, every computer terminal accessing the ILS will also be able to access all bibliographic services to which the library subscribes, and hopefully, through a User-Cordial Interface. This will not only allow ease of access to such national resources for the student or library patron, but also allow the library to aggregate data and effectively manage this information resource.

The ability to access such external systems from an LRC is clearly dependent on overall system design and integration. Online information systems could play a very important role in the total educational process. An author that realizes the breadth of the LRC resources can develop interactive courseware based on other information systems available to his or her student. The student can more effectively learn to use resources which will become part of his or her professional environment. Other network resources are available to the author/student in this LRC; however, the two presented should provide the reader with an appreciation for the potential of the resource sharing in a learning environment. Additional examples may only detract from the primary point; that is, an LRC should be a complete Learning Resource Center, not constrained by either walls or a particular computer system.

VI. CONCLUSIONS AND FUTURE DIRECTION

The above paragraphs provide a brief glimpse into the present and future potentials for Learning Resource Centers. Many of the references clearly show the present direction, both in CBEM and Information Systems. Still lacking, however, is the integration of a complete system for demonstration. The research program at LHNCBC in Advanced Information Systems, Computer-based Educational Systems, and Computer Technology are all designed to help meet that goal.

REFERENCES

1. Smith, Stanley G., and Bruce Arne Sherwood, "Educational Uses of the PLATO Computer System," Science, Vol. 192, pps. 344-352, April 23, 1976.

2. Capece, Raymond P., et al., "Technology Update," Electronics, Vol. 51, No. 22, pps. 110-221, October 26, 1978.

3. Chapman, Glen I. and William H. Ford, "LHC-PILOT System Managers Guide," LHNCBC Technical Report No. TR-7802, May 1978.

4. Chapman, Glen, I. and William H. Ford, "LHC-PILOT Users Guide," LHNCBC Technical Report No. TR-7801, May 1978.

5. Starkweather, John A., "Guide to 8080 PILOT, Version 1.1," Lister Hill National Center for Biomedical Communications Internal Report. Also NTIS PB-270 715/6WC.

6. Goldstein, Charles M. and William H. Ford., "The Lister Hill Center Plasma Terminal System and Implications for Medical Networks," Proceedings of the Association for the Development of Computer-based Information Systems, March 1978.

7. Coursewriter is an IBM product. Refer to IBM for latest manual and documentation.

8. Stifle, Jack, "A Glimpse of the PLATO V Terminal," In New Directions in Educational Computing: Proceedings/1977 ADCIS Winter Conference in Wilmington, Delaware, pps. 186-191.

9. "American National Standards MUMPS Language Standard," MUMPS Development Committee, American National Standards Institute, Inc. September 15, 1977.

10. Zimmerman, Joan, "Book of MUMPS," MUMPS Users Group, 700 South Euclid Avenue, St. Louis, Missouri 63110, U.S.A., 1977.

11. Lister Hill National Center for Biomedical Communications, "Intelligent Device Controller," Internal LHNCBC Document, 1978.

12. Kenney, George C., "Special Purpose Applications of the Philips - MCA Videodisc System," IEEE Consumer Electronics Transactions, published by the IEEE, November 1976.

13. Lister Hill National Center for Biomedical Communications, "Videodisc Program Plan," Internal LHNCBC Document, 1978.

14. Lister Hill National Center for Biomedical Communications, "Advanced Terminal System Users Guide," Internal LHNCBC Document, 1978.

15. Schoolman, Harold M., and Lionel M. Bernstein, "Computer Use in Diagnosis, Prognosis, and Therapy," Science, Vol. 200, No. 4344, pps. 926-931, May 26, 1978.

16. Lister Hill National Center for Biomedical Communications," The Integrated Library System," Internal LHNCBC Document, 1978.

17. Ford, William H. and Kristin N. Johnson, "Microprocessor Implementation for Full Text Data Base With Graphics," Proceedings of the 1978 MUMPS Users' Group Meeting, June 1978.

18. Goldstein, Charles M. and William H. Ford, "The User Cordial Interface," Online Review, Vol.2, No. 3, pps. 269-275, 1978.

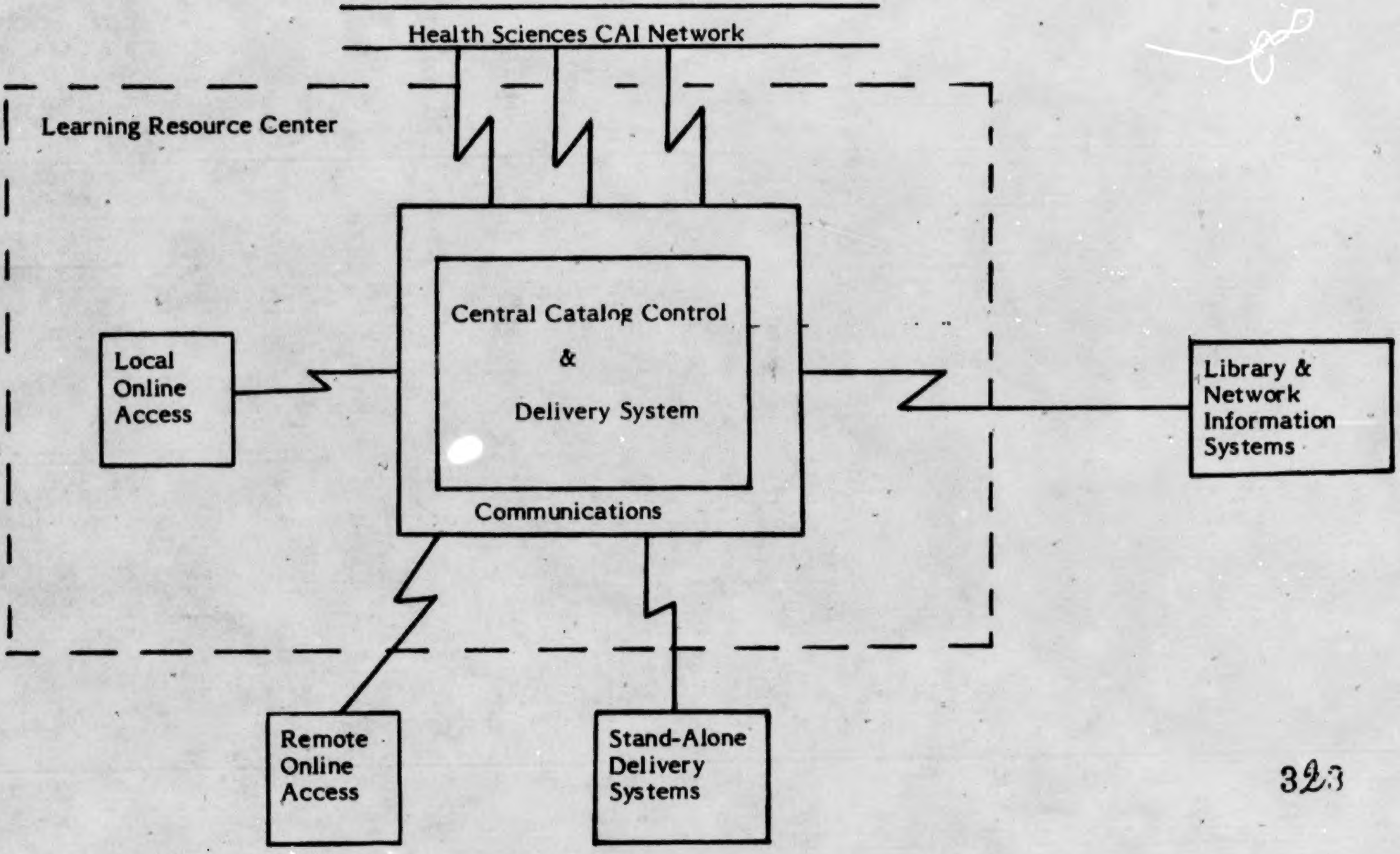

Health Sciences CAI Network
Learning Resource Center
Central Catalog Control
&
Delivery System
Communications
Local Online Access
Library & Network Information Systems
Remote Online Access
Stand-Alone Delivery Systems

Figure 1. A Model Learning Resource Center (LRC)

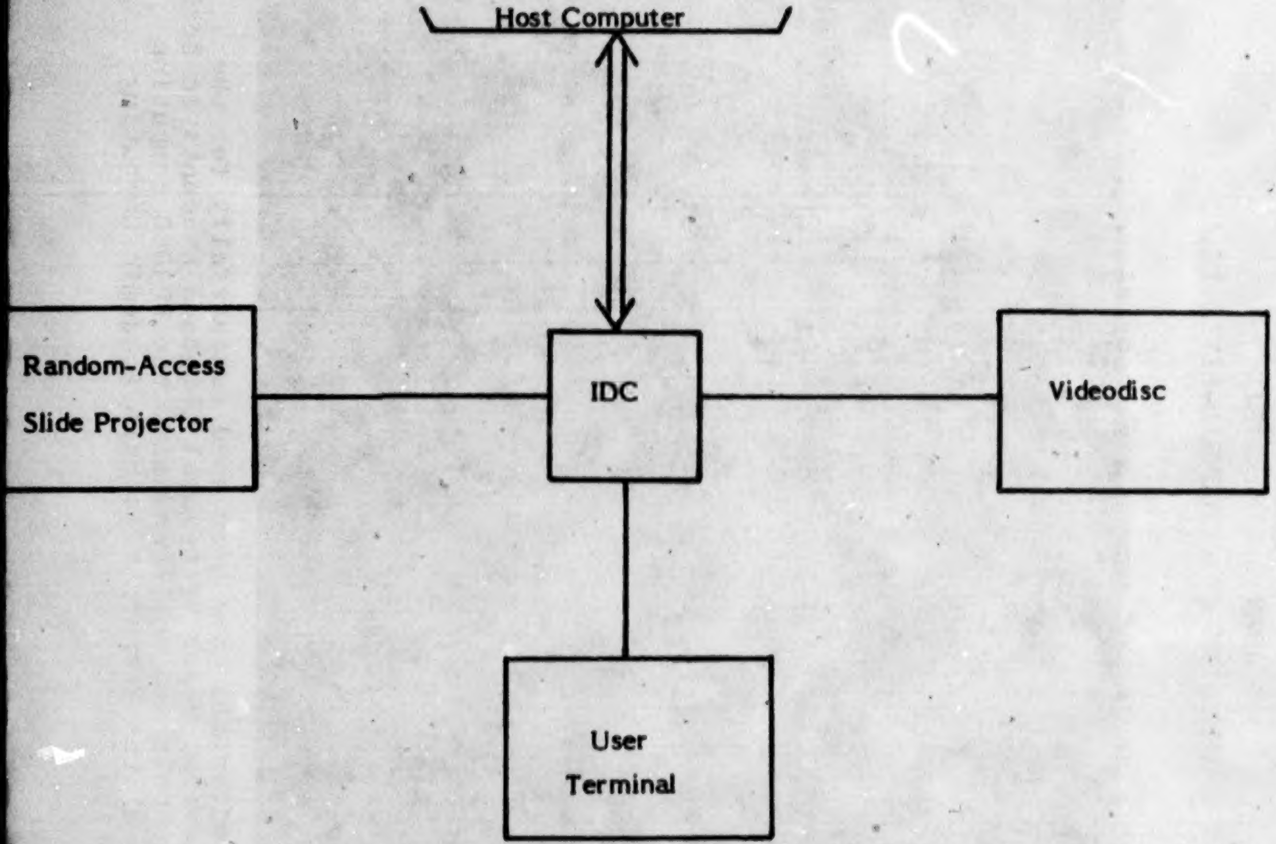

Host Computer
Random-Access
Slide Projector
IDC
Videodisc
User
Terminal

Figure 2. Intelligent Device Controller (IDC)

SELF-PACED COMPUTER CURRICULUM PROJECT
BASIC AND COBOL

DAVID L. FEINSTEIN
AND
JEAN L. FEINSTEIN

UNIVERSITY OF WISCONSIN-RIVER FALLS

Through a series of summer workshops since 1972, the University of Wisconsin-River Falls has developed self-paced courses for instruction in BASIC and COBOL. The courses are the equivalent of the Introduction to Computing and Introduction to Business Data Processing classes taught at the University.

The BASIC materials consist of about one hundred individual modules. These modules are organized into ten units covering the various elements of BASIC programming. The first unit includes the PRINT, LET, INPUT, READ-DATA, GO TO, IF-THEN and FOR-NEXT statements. More advanced topics such as arrays and files are in individual units.

Each module introduces a new concept and has as the key component a BASIC program. The student is expected to "GET" the program and, through experimentation, answer certain questions. A manual/textbook has been written which includes the objective, listing, execution, explanation and questions for each module. The COBOL modules follow the same general format.

The format of the presentation has varied considerably. At the University, the material is presented in a large lecture hall equipped with TV monitors with on-line access to the computing system. At the remote sites, the students proceed at their own pace. Formal registration is through University Extension. There are eight tasks for each course.

These materials have been used successfully for the past several years by several thousand students in both high schools and colleges using the HP3000 computing system at the Western Wisconsin Academic Computing Consortium (WACC).

EXPERIMENT IN REMOTED INSTRUCTION FOR THE FEDERAL AVIATION ADMINISTRATION DEPARTMENT OF TRANSPORTATION

BY

THELMA STACY AND CRAIG BURSON

The Federal Aviation Administration (FAA) is conducting a test and evaluation project to determine if it is feasible to remote technical training to certain of its field facilities. In the project a course on Automatic Mode Jet Flight Inspection Procedures is being taught in each of its seven continental U.S. Flight Inspection Field Offices using PLATO.

Formerly, this course was taught in-residence for two weeks at the FAA Academy in Oklahoma City, Oklahoma. It was completely individualized under the PLATO computer-managed instruction format, eliminating the need for students to travel to the FAA Academy. Various learning resources were developed to substitute for classroom lectures. These resources included manuals, workbooks, slide cassette packages, video tapes, and computer-assisted instruction (CAI).

The CAI portion included a complete operational simulation of the airborne computer which has recently been incorporated into the FAA's flight inspection aircraft. The CAI simulation is used for instruction as well as testing of student mastery.

Training began in July 1978 and, on the basis of preliminary assessments, is proceeding satisfactorily in all seven locations. The course will shortly be extended to the FAA's Flight Inspection Groups in Hawaii and Germany. It is felt that the wide time zone variances will not be a problem because current student interaction with on-call instructors in Oklahoma City has been low.

QUICK/FTU

A Quasi-Interactive Computer-Based Instructional System at Florida Technological University

Principal Investigator: Dr. Larry K. Cottrell, Assistant Professor of Computer Science, Department of Computer Science, Florida Technological University, P. O. Box 25000, Orlando, Florida 32816. Phone (305-275-2341)

The CMI project at Florida Technological University (FTU) is the result of merging two CMI systems which were developed and tested at Purdue University in the 1974-1977 time period. The QUICK system developed by Dr. Sharon Fletcher and Dr. Thomas Luce and a CMI system developed by this investigator were written in FORTRAN. These sytems were implemented for test purposes on a CDC 6500 series computer at Purdue. The merged CMI system at FTU has been implemented on an IBM 370/165 computer.

A review of the literature on CMI reveals that most CMI systems do not permit students to directly interact with the system. Both input and output to these CMI systems are usually handled by teacher, teacher assistants, operators, etc. The QUICK/FTU system permits direct student interaction. A set of simplified commands exists for the students using this system. These commands allow students to: 1) REQUEST lesson content; 2) ANSWER questions and problems; and 3) INQUIRE about their progress throughout a lesson. Typically, after keypunching the desired commands on cards, the student has the input read by a self-service card reader and will wait a few minutes for the output to be printed by a high-speed line printer. All job submitted by the users of the system are processed in a batch mode. Hence, the system also accommodates a problem that is common to computer-based instructional systems which is variable student response time.

By using the QUICK/FTU system, an instructor has the opportunity of constructing unique lessons for each member of a class. Lessons are constructed by specifying a subset of TASKS from a given set of TASKS that are resident in the system. A TASK is the basic unit of instruction for the QUICK/FTU system and may contain, in addition to instructional content, up to 30 questions which are to be answered by the student. The student completes a lesson by passing a predetermined percentage of the TASKS which make up his lesson. A TASK is completed by completing the assignment and the questions associated with the TASK.

The QUICK/FTU system is currently being used in an experimental mode at FTU in order to determine its cost effectiveness and its impact on the host computing facility for a large number of users.

The Development and Utilization of Computer-Assisted-Instruction in an Activity-Based Introductory College Level Science Course

Martin F. Rudnick, Ed.D.
University of Northern Iowa
Cedar Falls, Iowa 50613

In 1977 the University of Northern Iowa received a grant from the National Science Foundation through the Comprehensive Assistance to Undergraduate Science Education program to develop an activity-based introductory science course. In the pilot year 1978-1979, sixty students are enrolled in this interdisciplinary course designed for non-science majors.

One componenet of our multifaceted teaching approach stressing student involvement is computer-assisted-instruction. This project report will discuss the development and utilization of graphic-augmented computer-assisted-instruction with college students having no previous experience with computers.

The initial step in the development process was evaluating the hardware on the market and determining what capabilities were essential given our budget and central computer (a Hewlett Packard 2000 access micro-computer). The next step was the development of software systems to administer graphic-augmented CAI. We decided to use Course Writer Facility for the majority of our course development because of its flexibility in branching to BASIC for the generation of graphics and for the storage of student responses. Secondly, we developed BASIC programs which would allow us to combine graphics input from multiple sources (digitizer, central computer, and terminal) into useful instructional visuals. Thirdly, we developed computer programs which would not only enable us to evaluate both formatively and summatively the effectiveness of the CAI, but to assess the progress of individual students as well. Finally, three programs-an introductory lesson on the use of the terminal, a classification program, and a program dealing with energy were developed for use in the first semester.

In 1978-80 school years, programs already developed will be administered and evaluated. Additional CAI dialogues will also be written. A computer-managed instruction facility will be developed to assist in the administration and evaluation of instruction for the larger number of students expected in subsequent semesters.

MICROCOMPUTER CONSTRUCTION IN HIGH SCHOOL

by

Paul Sadler, Sehome High School, Bellingham, WA

MARINER I, the student built computer at Sehome High School, Bellingham, WA is a Processor Technology SOL-20 microcomputer an 8080A processor chip. Twenty-seven students, selected by two instructors, built the computer from a kit during fall quarter 1977 as part of a "Computer Design and Construction" class. The students ranged in math ability from Practical Mathematics through Calculus and in age from freshmen to senior. The selection criteria was motivation, ability in electronics, and achievement in Computer Science courses. Two instructors were required by the necessity for attention to detail, need for expertise in electronics, and knowledge in the computer science field. Jack Crandall teaches Physics and has developed an intensive digital electronics course. Paul Sadler teaches a computer literacy course and computer programming courses in a variety of languages.

The Computer Design and Construction Class met M,W,F for 2 hours after school to eliminate conflicts with other elective courses. Our immediate objectives were to teach electronics, construction techniques, and soldering skills. Students worked in groups of 3 to learn the theory of operation and construct an 8 transistor radio kit. Soldering skills were perfected during this first phase and the quality of the finished computer and lack of construction errors proved this time was well spent.

Phase 2 contained three concurrent sessions; 1. digital electronics utilizing a circuit designer, 2. construction of the computer components, and 3. machine language (hex) programming. The three groups rotated each 2 weeks to allow all students to build a portion of the computer. All construction work took place in teams of 3 students. One student would read the instructions, a second student did the soldering, and the third student would inspect the results. This team work and double checking prevented any construction errors.

The Computer Design and Construction Class was a big success with the entire faculty and student body taking pride in our accomplishments. Our plans are to have as many students as possible use the computer and to DO IT AGAIN !

PROJECT REPORT

A COMPUTER-ASSISTED COMPUTER COURSE FOR ELEMENTARY AND MIDDLE-SCHOOL STUDENTS

Norman T. Bell
Michigan State University
East Lansing, MI 48824

Lee Droegemueller
Independent School District #196
Rosemount, MN 55068

For the past two years, there has been in operation in the elementary and middle schools of Independent School District #196 a computer course utilizing the capabilities of the low cost microcomputers. The thrust of this course has been to provide computer literacy at varying levels of complexity for more than 2,000 students in grades 4 through 8.

Three characteristics of this course appear to have caused it to be successful. First, the teaching of a computer course in grades 4 through 8 is a timely subject. Never before has so much computing power been available at such a low cost. Consequently, the schools are finding that many students are learning much about computing outside the classroom, and a large proportion of the remaining students appear to be very interested. Thus, computers are a real point of interest for students.

A second characteristic of the course is that computers are used to assist in teaching. Since the availability of computers has so increased, and costs decreased, every classroom has in it a microsystem to which students have access during the course. All concepts taught are first demonstrated, and then practiced by the students. Thus, the computer is used to learn about computing.

A third characteristic of the course is that a complete set of teaching materials has been developed for use by the teachers. Included in this set are overhead transparencies, student worksheets, and a detailed set of teacher instructions contained in a teacher guide. So complete are these materials that it is easier for a teacher to use them and be guaranteed of success than to ignore them and reduce the chances of success.

Thus, a highly successful program has been designed and implemented to teach an exciting topic to students using computer assistance.

CAI in Pharmacology at the University of Alberta.

Louis A. Pagliaro, M.S., Pharm.D., Charles M. Bidwell, Ph.D., David A. Cook, M.A., D.Phil., Charles Heath, Ph.D. (Faculties of Pharmacy and Medicine, University of Alberta, Edmonton, ALTA T6G 2H7 Canada).

A three year project designed to develop and test the suitability of computer-assisted instruction (CAI) in the teaching of pharmacology is being funded by Alberta Advanced Education and Manpower. The general purpose of this project is to:A. evaluate the learning which occurs when computer-based techniques are applied in pharmacology, B. determine those aspects of pharmacology which are best taught in this manner, and C. compare the results obtained with those obtained earlier using other teaching methods. Phase I of the project (2/78 to 12/78) consisted of adaptation of computer courseware obtained from the CATS consortium (University of Kansas), and the writing of new computer lessons. Thirty-eight lesson-programs have been developed and tested with a student population. Responses to a student questionnaire and usage statistics confirm our impression that CAI is perceived by the student as a valuable learning resource. We are not yet in a position to assess objectively whether student performance improved as a result of the time spent by the student. This aspect is being examined during Phase II (9/78 to 9/79) in which the courseware is being student-tested and revised as needed. During Phase III (1/79 to 1/81) student trails will be conducted over a two-year period in all pharmacology undergraduate courses. At all stages of the project data will be collected to assess student learning, student use of, and attitudes towards the CAI lessons. Teaching faculty will be monitored with regard both to attitude to CAI and to program development and updating. A further aim of this project is to examine delivery systems for CAI. Preliminary investigation of the use of the SOL II microcomputer with programs written in PILOT language are encouraging. If these expectations are realised, the prohibitive costs of CAI, particularly for small centers without a large computer, may be reduced greatly. This has important implications for continuing education in medicine and in pharmacy. Many questions to which we can now only provide speculative answers will be answered definitively by studies to be carried out during the remainder of this project.

"Why Computer Commitment Varies Among High School Teachers"

Ronald E. Anderson, Thomas Hanson, David C. Johnson, Daniel Klassen, Minnesota Educational Computing Consortium

Previous research has found instructors as well as students ambivalent in their attitude toward instructional computing. Some teachers embrace the computer with open arms using it in many ways; other teachers stand off and say "show me that it isn't a waste of my time." Feelings range from excitement to disinterest, from addiction to anxiety.

Commitment has two facets: acceptance and involvement. While a teacher might accept or adopt the computer because of administrative decree, intense involvement depends upon a highly favorable attitude toward computers. Thus attitude and commitment are closely interrelated. Our challenge is to find out what accounts for variation in teacher attitude and commitment to computing.

A 1977 survey of mathematics, science, and business education teachers in Minnesota secondary schools was designed with this question in mind. Questions were asked about a variety of classroom computer activities and a measure of computer commitment was constructed from these. Regression analysis of over 1,000 teachers reveals that convenient availability of computer facilities is critical to computer commitment. Teacher training in computing is also a major factor. Other variables including attitudes toward educational computing make it possible to explain much of the variation in teacher commitment to computers.